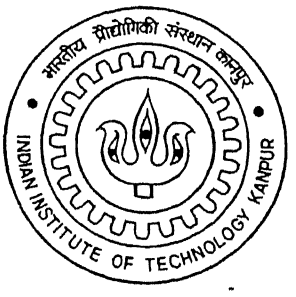
OPTIMAL DESIGN OF PSEUDO-INTRINSIC SURFACES USING GENETIC ALGORITHM TECHNIQUE

by

B. KUMARA SWAMY



DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
September, 1997

OPTIMAL DESIGN OF PSEUDO-INTRINSIC SURFACES USING GENETIC ALGORITHM TECHNIQUE

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

B. KUMARA SWAMY

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

September, 1997

ME- 1997-M-SWA-CPT

In fond memory of

my ever loving sister late Smt. RAJA MANI

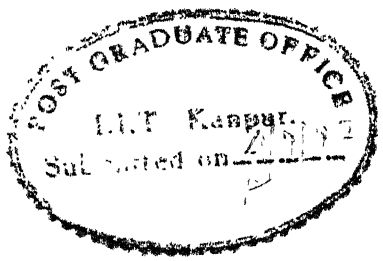
CERTIFICATE

It is certified that the work contained in the thesis entitled **Optimal Design of Pseudo-Intrinsic Surfaces Using Genetic Algorithm Technique** by **B. Kumara Swamy**, *has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.*

September, 1997

Dr. Sanjay G. Dhande
Professor, ME and CSE
IIT Kanpur

Dr. Kalyanmoy Deb
Assoc. Professor, ME
IIT Kanpur



ACKNOWLEDGEMENTS

I would like to express my deep sense of gratitude to my ever loving guides, Dr. Sanjay G. Dhande and Dr. Kalyanmoy Deb, for their invaluable guidance and help throughout my M. Tech programme. I am sincerely thankful for their valuable suggestions in my academic as well as personal life.

I am paying great regards to Prof. S. G. Dhande for sparing his precious time in giving valuable hints and ideas which have helped me to accomplish this master's thesis successfully. I am extremely grateful for his constant encouragement and motivation during my programme. It would not be an exaggeration to say that my experiences with him tempt me to pursue my remaining career with him.

I am immensely thankful to my beloved wife Kavitha for her moral support and constant encouragement. I could find no words to express her greatness in understanding and sharing my feelings as my better-half. I express my sincere regards to my parents and brother for their cooperation and valuable help in times of need for the successful completion of my M. Tech programme.

I am grateful to Dr. Gautam Biswas for the useful discussions I had with him regarding my thesis work. I am thankful to my CAD-P friends Krishna Moorthy, Ugandhar, Yogesh, Tarun, Rajarathinam, Siddharth, Awadhesh and others for their help and pleasant company. I also thank Mr. A. D. Bhatt, Mr. Sanath Agrawal, Mr. C. P. Singh, and Mr. S. G. Gupta for their help in one way or the other.

My thanks are also due to my family friend Mr. N. Dastagiri Reddy and his family for their affection. I would like to thank my friends Mr. P. S. Rao, Mr. N. Chandra Shekhar and many others who have made my stay at IIT Kanpur a pleasant and memorable one.

Finally I am grateful to the Almighty for what I am today.

- B. Kumara Swamy

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

a	Vector of nodal degrees of freedom
b	Binormal vector of a generic point P
B	Strain-displacement matrix
D	Material matrix
E	Young's modulus
f	Right side vector
F(X)	Objective function
G	Maximum number of generations
g(X)	Inequality constraint
h(X)	Equality constraint
J	Jacobian matrix
K	Stiffness matrix
m	Slope of the tangent vector
N	Population size
N	Shape function matrix
p_c	Crossover probability
p_m	Mutation probability
r	Position vector of a generic point
s	Arc length from a reference point to a generic point measured along the curve
t	Tangent vector of a generic point

t	Thickness of the shell surface
T	Transformation matrix from local to global coordinate system
u, v, w	Nodal displacements
x, y, z	Cartesian coordinates of a generic point P
xyz	Global coordinate system
$x'y'z'$	Local coordinate system
X	Vector of design variables
x_i	design variable

Greek Symbols

Δ	Area under the κ–s plot
α	change in tangent angle
κ	Curvature of a generic point P
λ	Sharpness of the clothoid pair
ν	Poisson's ratio
τ	Torsion of a generic point
ξ	A parameter varying from 0 to 1
ψ	Tangent angle
θ	Nodal angular degree of freedom

ABSTRACT

In the present work, a methodology of shape optimization has been developed. For any shape optimization process, it is necessary to have a module for shape synthesis. In the present work, the shape synthesis is accomplished by means of a pseudo-intrinsic definition of a surface patch. This definition provides fewer shape design variables and also allows direct control over the higher order properties such as curvature. The shape synthesis method includes mapping of points of a surface from the intrinsic space to the Cartesian space.

The present work uses the technique of Genetic Algorithm for carrying the process of optimization. It has been found that this technique allows realization of optimal as well as sub-optimal surfaces. The GA method has been found to be computationally effective for shape optimization problems. The design of an optimal shape can be verified by making a model from a rapid prototyping process. This has been also accomplished in the present work. The proposed methodology is illustrated with the help of two case studies. The first example is of a minimum area surface. The second problem is of maximizing the stiffness of a shell surface.

Chapter 1

INTRODUCTION

1.1 Introduction

Shape optimization is a kind of problem which is currently attracting a significant level of interest of engineers and scientists, within the broad field of optimization of mechanical and structural elements. Different reasons can be attributed for this increasing interest: the development of general and powerful techniques for the analysis of complex shapes (e.g. Finite elements, Boundary elements); the appearance of new and efficient optimization algorithms for linear and nonlinear optimization problems with a great variety of constraints; and the use of experimental techniques that allow checking of results. This growing interest in shape optimal design reflects a realization of shape changes for improving structural performance. Shape optimization of a component can be defined as deciding the geometry of a curve or a surface which will maximize or minimize an objective function as well as satisfy a set of constraints. For many problems shape design is more effective than the sizing or cross section design [22]. A typical example is that of a stress concentration around a hole boundary in a plate. Sizing optimization would increase or decrease the thickness of the plate near the hole, while shape design would change the shape of the hole boundary itself.

Even though there are a number of numerical optimization methods [44] which have been used in various shape and other optimization problems, Genetic algorithm (GA) is relatively a new field of optimization which is based on principles of natural genetics. Genetic algorithms are probabilistic optimization methods that work on a population of designs by recombining desirable features of existing designs. In the last decade, GAs have proven their ability to deal with a large

class of problems. It has been found that in structural optimization, genetic approach appears especially promising in case of large, nonconvex, integer programming problems [9]. In the present research work, it is proposed to apply GA to obtain optimal surface designs with respect to two objective functions like surface area and stiffness.

1.2 Surface Design and Applications

Three dimensional or space curves and surfaces play an important role in engineering, design, and manufacturing of a diverse range of products, e.g. automobiles, ship hulls, aircraft fuselages and wings, propeller blades, shoes, bottles, buildings etc. They also play an important role in the description and interpretation of physical phenomena, e.g. in geology, physics, and medical sciences. Prior to the development of mathematical and computer models to support engineering, design, and manufacturing, descriptive geometry was used. Many of these geometric design techniques have been carried over into the computer aided geometric design.

Surfaces and their description has been playing a critical role in the fields of engineering, design, prototyping, and manufacturing. The design and manufacture of automobile bodies, ship hulls; glassware and bottles; aircraft fuselages and wings; propeller, turbine, compressor and fan blades; furniture, and shoes are some of the obvious examples. Other examples include, basket and soccer balls, funnels, beer cans, and satellite antennas. Surface shape or geometry is the essence of design for either functional or aesthetic reasons. Surface description also plays an important role in the representation of the data obtained from medical, geological, physical and other natural phenomena.

In the recent past, the design of car bodies and scooters, for racing especially, have been undergoing drastic improvements in order to reduce the drag resistance and to improve the overall performance of these vehicles. Design of passenger cars and bus bodies have also been continuously changing over the years to not only increase the efficiency of the engine but also to improve the aesthetic appearance of theses vehicles. There has been tremendous research in

progress to develop or come up with an aerodynamically smooth surface or shape designs for better and efficient performances in automobile and aerospace industries.

1.3 Review of Surface Design using CAD and Optimization

Computer Aided Design (CAD) may be defined as any use of computer in the design of an individual part, a subsystem, or a total system [36]. The use does not have to involve graphics. The design process may be at the system concept level or at the detailed part design level. It may also involve an interface with Computer Aided Manufacturing (CAM). CAD can also be defined as the use of computer systems to assist in the creation, modification, analysis, or optimization of a design.

In design and engineering the traditional way of representing a surface is to use multiple orthogonal projections. In effect the surface is defined by a net or mesh of orthogonal plane curves lying in plane sections plus multiple orthogonal projection of certain three dimensional feature, or detail lines. The clay stylist's model traditionally used is an example for this type of surface representation.

In computer graphics and computer aided design, it is advantageous to develop a *true* 3-D mathematical model of a surface. Such a model allows early and relatively easy analysis of surface characteristics e.g., curvature, torsion, or of physical quantities that depend upon the surface e.g., volume, surface area, mass moment of inertia etc. Visual rendering of the surface for design or design verification is simplified. Further, generation and manipulation of the necessary information required to fabricate the surface, e.g., numerical control codes, is also considerably simplified as compared to the traditional net of lines approach.

There are two basic philosophies embedded in the surface description techniques [36]. The first, mostly associated with the name of Coons, seeks to create a mathematical surface from known data. The second, mostly associated with the name of Bezier, seeks to create a mathematical model of a surface *ab initio*. Initially, engineering discipline which depends on numerical parameters was attracted towards the Coons approach, while disciplines that depend

upon visual, tactile, or aesthetic factors, e.g., stylists and graphic artists, were attracted towards the *ab initio* (Bezier) techniques. However, recent works by Rogers with real time interactive systems for design of ship hulls and by Cohen for general surface design shows that these two approaches are compatible.

Definition of a space curve is a must for any surface design. There are numerous techniques by which a curve can be generated ([36], [27], and [16]). These techniques are broadly classified into *curve fitting* and *curve fairing* techniques. Curve fitting techniques are characterized by the fact that the derived curve passes through each and every data point. e.g., Cubic spline and Parabolically blended curves. But, in curve fairing techniques the mathematical description of a space curve is generated *ab initio*, i.e. without any prior knowledge of the curve shape or form. e.g., Bezier and B-spline curves. Of late, rational curve and surface description is well known in the literature. e.g., Non-Uniform Rational B-splines (NURBS). They provide a single precise mathematical form capable of representing the common analytical shapes–lines, planes, conic curves including circles, free-form curves, quadric and sculptured surfaces used in computer graphics and CAD.

Surface Design

There are various techniques to generate a three dimensional surface by using any of the above mentioned curves, viz., surfaces of revolution, sweep surfaces, Coons' bicubic surface, Bezier, B-spline, and NURBS surfaces; lofting and sculptured surfaces etc. Surfaces can also be designed using parametric and non-parametric representation. Now we will review some of these techniques in the following discussion.

Surfaces of revolution: Perhaps, the simplest method for generating a 3-D surface is to revolve a 2-D entity, e.g., a line or a plane curve, about an axis in space. e.g., Revolving a line in the XY-plane about the X-axis results in a cylindrical surface. Other examples include conical surface, sphere, ellipsoid, torous, paraboloid, and hyperboloid etc. A B-spline surface of revolution is shown in Figure 1.1.

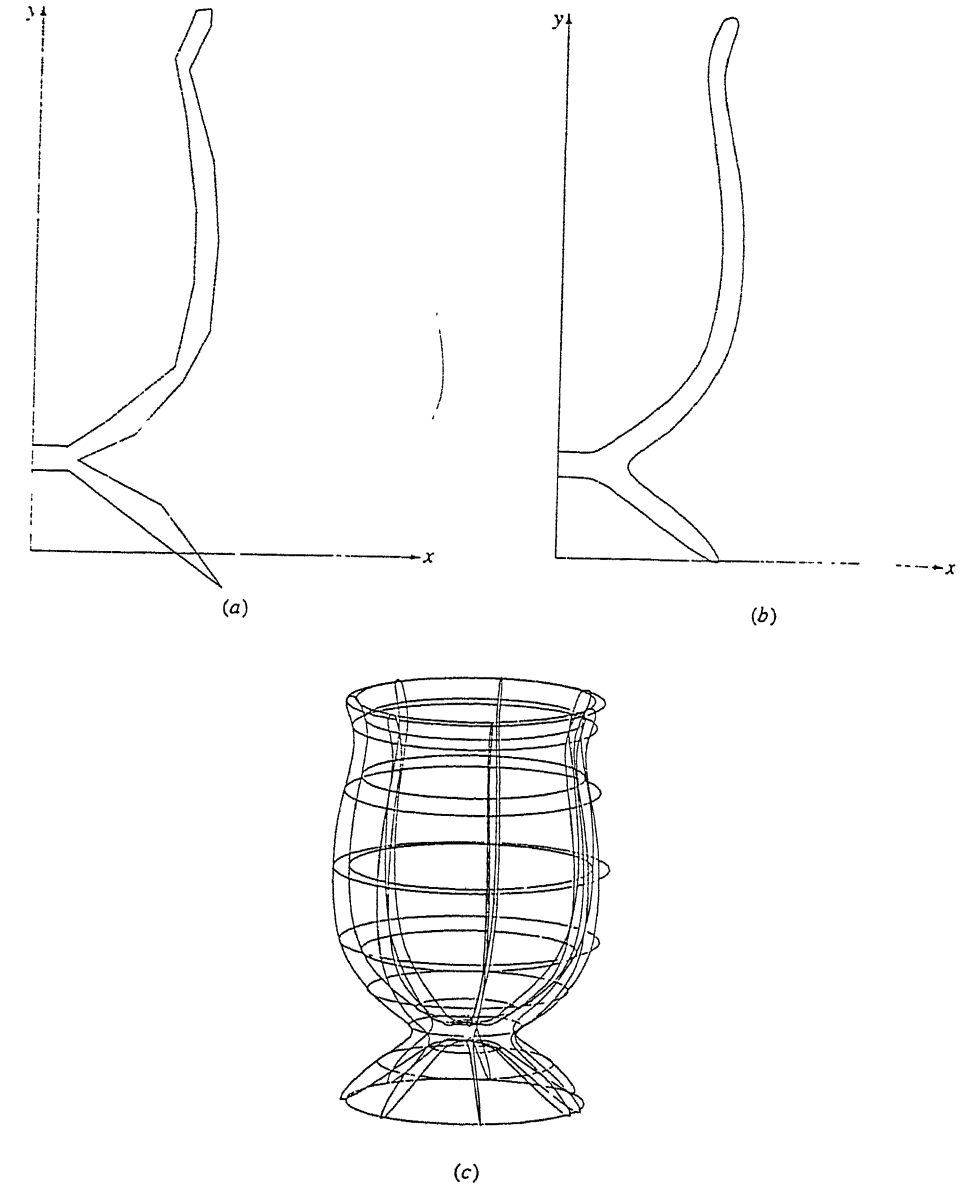


Figure 1.1: A B-spline surface of revolution. (a) Polygon vertices; (b) B-spline curve; (c) surface [36]

Sweep surfaces: When a line, polygon, or a curve is traversed along a path in space, the resulting 3-D surfaces are called sweep surfaces. Sweep surface generation is frequently used in geometric modelling. Figure 1.2 shows a sweep surface generated from a single cubic spline curve segment swept parallel to the Z-axis.

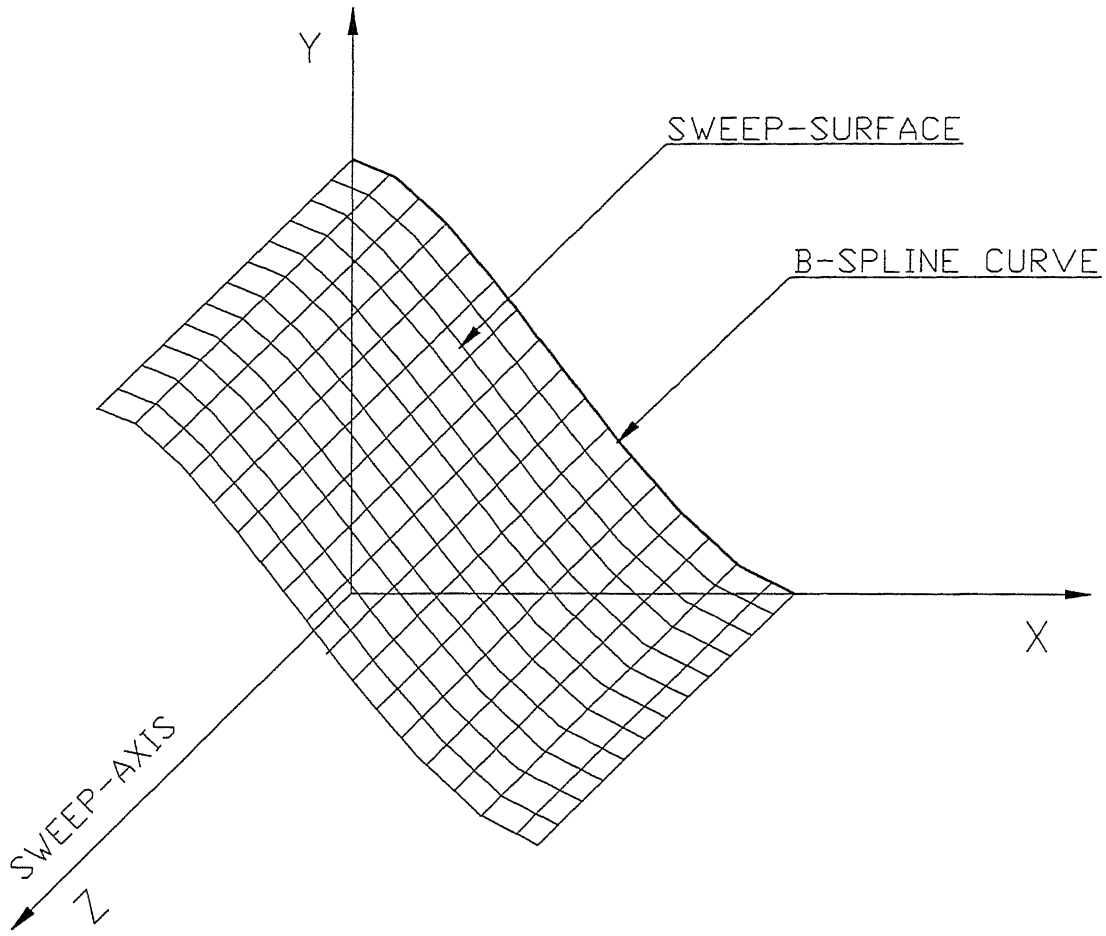


Figure 1.2: A cubic spline based sweep surface.

Surfaces of general form: These include Coons' bicubic surface, Bezier and B-spline surfaces etc. The **Coons' bicubic surface** patch uses normalized cubic splines for each of the four boundary curves [36]. Cubic blending functions are used to define the interior of the patch. Coons' bicubic surfaces provide a flexible and powerful surface design tool. However, as with the cubic spline curves, practical usage suffers from the necessity of specifying precise non-intuitive mathematical information, e.g., position, tangent and twist vectors. Most of the difficulties associated with Coons' bicubic surfaces are overcome with **Bezier surfaces**. Each boundary curve of Bezier patch is a Bezier curve. Bernestein basis is used for the surface blending functions. Some of the properties of the Bezier surfaces are given below:

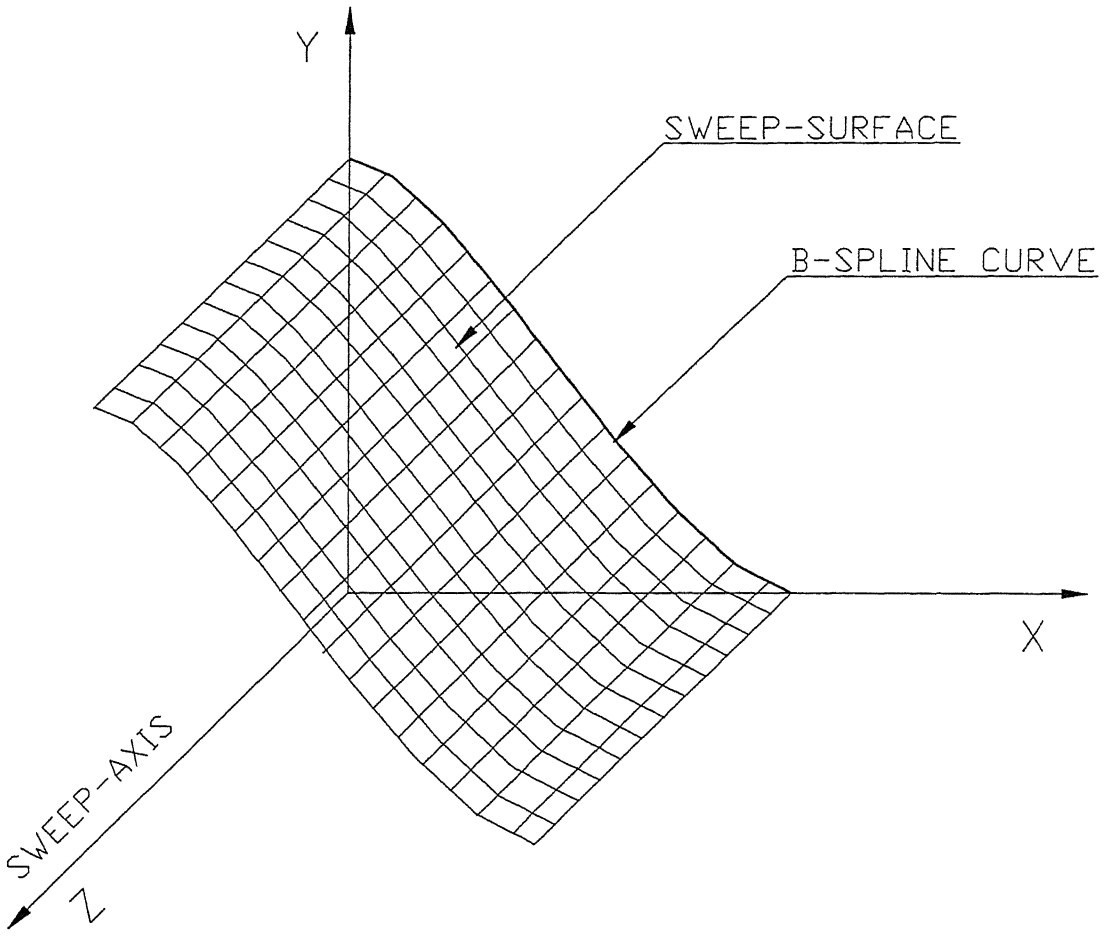


Figure 1.2: A cubic spline based sweep surface.

Surfaces of general form: These include Coons' bicubic surface, Bezier and B-spline surfaces etc. The **Coons' bicubic surface** patch uses normalized cubic splines for each of the four boundary curves [36]. Cubic blending functions are used to define the interior of the patch. Coons' bicubic surfaces provide a flexible and powerful surface design tool. However, as with the cubic spline curves, practical usage suffers from the necessity of specifying precise non-intuitive mathematical information, e.g., position, tangent and twist vectors. Most of the difficulties associated with Coons' bicubic surfaces are overcome with **Bezier surfaces**. Each boundary curve of Bezier patch is a Bezier curve. Bernestein basis is used for the surface blending functions. Some of the properties of the Bezier surfaces are given below:

- The degree of the surface in each parametric direction is one less than the number of defining polygon vectors in that direction.
- The continuity of the surface in each parametric direction is two less than the number of defining polygon vectors in that direction.
- The surface generally follows the shape of the defining polygon net.
- The surface is contained within the convex hull of the defining polygon net.
- The surface is invariant under an affine transformation. i.e., the surface is transformed by transforming the defining polygon net.

B-spline surfaces are a natural extension of Bezier surfaces, wherein B-spline basis functions are used instead of the Bernestein basis functions to describe both the boundary curves and to blend the interior of the surface. B-spline surfaces are the most widely used free-form surfaces. Currently they are mainly used for *shape representation*, but not as widely and directly for *shape synthesis* [46]. In the context of *shape representation*, only geometric data, and no underlying physical constraints are considered. The use of B-spline surfaces for this is a pure mathematical representation which has many advantages such as geometrical intuitiveness, rich representation capability, and data reduction. But in *shape synthesis* or *shape design*, geometric shapes are intended to serve certain functions. They are designed according to their underlying physical constraints such as aerodynamic and hydrodynamic constraints. Important properties of B-spline surfaces are listed below [36]:

- The maximum order of the surface in each parametric direction is equal to the number of defining polygon vectors in that direction.
- The continuity of the surface in each parametric direction is two less than the order in each direction.
- The surface lies within the convex hull of the defining polygon net.
- The surface is invariant under an affine transformation.
- If the number of defining polygon vertices is equal to the order in each parametric direction and there are no interior knot values, then the B spline surface reduces to a Bezier surface.

Because of the convex hull properties of the B-spline curves, a B-spline surface can contain imbedded flat regions and lines of sharp discontinuity, which is a particularly desirable characteristic for many design situations. B-spline surfaces have excellent local control properties which make them attractive for many of the design activities in engineering, design, and manufacturing. Figure 1.3 shows a third order (in each parametric direction) open B-spline surface and its defining polygon net.

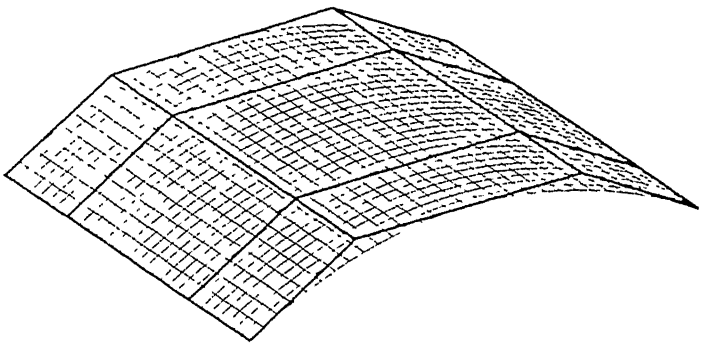


Figure 1.3: A third order open B-spline surface [36]

The latest development in surface design using CAD is the use of rational curves. Of these, Non-Uniform Rational B-spline (NURBS) surface has received special attention because of their ability to represent quadric surfaces and to blend them smoothly into higher degree sculptured surfaces [36]. Both, complex free-form surfaces and standard analytic surfaces can be represented in a unified manner using NURBS [32]. A NURBS surface is generated using the NURBS curve for each of the boundary curves and to blend the interior of the surface patch. NURBS surfaces have similar analytical and geometrical properties to their non-rational counterparts. At present, NURBS have become the industry standard in computer aided geometric design (CAGD).

Recently, cyclide surface patches have been successfully used in piping and automotive industry [38]. Cyclide surfaces are found to have low algebraic degree, exact NURBS representat-ion and circular lines of curvature. Another development is RaG formulation. Geometric modelling systems based on Rational Gaussian (RaG) formulation have the unique ability (which is not shared by NURBS formulation) to design closed, half-closed, and open shapes from irregularly spaced points [17]. B-spline basis is replaced by gaussian frequency functions which enable the control of curvature values in the generated shape. However, rational gaussian surface is local and the computation of a surface point theoretically depends on all of its control points.

Most of the surface design techniques are based on control points. In order to alter the shape the control points need to be adjusted which involves many design variables. Also control of higher order properties like curvature, torsion etc. is not possible with the control point approach. On the other hand intrinsic definition of a shape involves less number of design variables and the intrinsic properties are found to relate some of the physical parameters like deflection of a beam etc.

Optimization Methods

The concept of optimization is basic to much of what we do in our daily lives. The desire to run faster a race, win a debate, or increase the efficiency of a machine implies a desire to do or be the best in some sense. In engineering, we wish to produce the "best quality of life with the available resources". Thus, for 'designing' new products, we must use design tools which provide the desired results in a timely and economical fashion. Optimization methods are the required design tools which are finding increasing popularity in engineering design activities both in industry and academia, partly due to the availability and affordability of computers. A number of software modules have been coming up in the field of optimization some of which have been incorporated in the well-known engineering analysis (FEM/BEM) pakages.

A typical nonlinear constrained optimization problem can be stated as follows [44]:

$$\text{Minimize:} \quad F(\mathbf{X}) \qquad \text{objective function} \tag{1.1}$$

Subject to:

$$g_j(\mathbf{X}) \le 0 \quad j = 1, m \qquad \text{inequality constraints} \tag{1.2}$$

$$h_k(\mathbf{X}) = 0 \quad k = 1, l \qquad \text{equality constraints} \tag{1.3}$$

$$\mathbf{X}_i^l \le \mathbf{X}_i \le \mathbf{X}_i^u \quad i = 1, n \qquad \text{side constraints} \tag{1.4}$$

where,

$$\mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{Bmatrix} \qquad \text{design variables} \tag{1.5}$$

The n-dimensional vector $\mathbf{X}$ is called as the vector of design variables. The objective function $F(\mathbf{X})$ as well as the constraint functions $g_j(\mathbf{X})$ and $h_k(\mathbf{X})$, may be linear or nonlinear functions of the design variable $\mathbf{X}$. These functions may be explicit or implicit in $\mathbf{X}$ and may be evaluated by any analytical or numerical technique we have at our disposal. Objective function $F(\mathbf{X})$ usually is the weight of a structure or deflection of a beam or overall cost involved or a combination of some objectives. Even though the above problem is of minimization, a maximization problem can be easily formulated using the duality principle which states that the problem of maximizing a constrained objective $F(\mathbf{X})$ is same as the problem of minimizing $-F(\mathbf{X})$.

If an optimization problem is simply stated as below:

$$\textit{find} \quad \mathbf{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T \quad \textit{which minimizes} \; F(\mathbf{X}) \tag{1.6}$$

then such problems are called unconstrained optimization problems.

Most of the traditional optimization algorithms can broadly be classified into two groups [8]: Direct search methods requiring only the objective function values and Gradient search methods requiring gradient information either exactly or numerically. The common similarity between of most of these traditional methods is that they all work by point by point basis.

Algorithm starts with an initial point (usually supplied by the user) and depending on the transition rule used in the algorithm a new point is determined. Essentially, these algorithms vary according the transition rule used to update the point.

Most of the traditional optimization techniques are suitable for well behaved, unimodal, simple objective functions.. Also most of these algorithms are not robust, because each of them is specialized to solve a special class of problems. On the other hand, Genetic Algorithms (GAs) are somewhat different search and optimization algorithms, which have been reportedly successful in solving a wide variety of search and optimization problems in sciences, engineering, and commerce. GAs are adaptive search and optimization algorithms that mimic the principles of natural genetics. Because of their simplicity, ease of operation, minimal requirements, and global perspective GAs have been used in a wide variety of problems [9].

It has been observed by many researchers that GAs result in a much better solution than a traditional optimization technique. The objective function and the constraint functions, if any, must be continuous and have continuous first derivatives in the design variables vector in order to use many of the traditional optimization methods [44]. As GAs work with only function values at various discrete points, a discontinuous or discrete function can be easily tackled by using GAs. The more striking difference between GAs, and most of the traditional optimization methods is that GAs work with a population of points instead of a single point. Thus a set of solutions can be obtained at every iteration (called a 'generation' in GA terminology) and it is very likely that the expected GA solution may be a global solution. GAs do not require any auxiliary information like gradient calculation except function values. Another difference between GAs and traditional techniques is that GAs use probabilistic rules to guide the search, whereas most of the traditional methods have fixed transition rules to move from one point to another point. That is why these methods, in general, can only be applied to a special class of problems, where any point in search space leads to the desired optimum. Since GAs use probabilistic rules and an initial random population, in the begining the search may proceed in any direction and when the population has converged in some locations the search direction narrows and a near optimal solution is found. Many of the engineering and other decision making problems have been solved using Genetic algorithms ([4], [10], [28] and [35]). These applications suggest that GAs can be used for a wide

range of problems with minimal requirements and yield better results than the traditional methods. In the present work we use this relatively new technique for the optimal design of pseudo-intrinsic surfaces.

1.4 Proposed Area of Research and Objectives

Shape is a basic property for design and manufacture, which can be defined as "those aspects of the geometrical form which have to do with the external aspect that an object presents to the world". Mathematical definition of a shape is very difficult. Shape represents a very important way with which we perceive and reason about the world. Shape is like an overall attribute or characteristic. It is a macroscopic property. Shape is an intrinsic property of geometry much like entropy is an intensive thermodynamic property[45]. Examples of shape intrinsic properties are the curvature and torsion of a space curve, gaussian and principal curvatures of a three-dimensional surface. Figure 1.4 shows the role of shape in a typical engineering design cycle. Shape is synthesized in the intrinsic domain using a mathematical model. This shape representation is then mapped into Cartesian domain, in which different analyses are performed based upon the particular application. The results are evaluated and updated till the optimum shape is found.

The method of shape optimization, proposed in this work, consists of selecting a shape model in an intrinsic space, defining a set of shape design variables (SDVs) and then mapping into the Cartesian space. A shape model is represented as a set of LINear Curvature Elements (LINCEs). The shape design variables are the values of curvature and/or tangent angles at some of the LINCEs. The overall objective of this work is to develop a CAD approach to shape optimal design of pseudo-intrinsic surfaces using genetic algorithms and rapid prototyping techniques and to illustrate how this methodology can be applied to some of the engineering design problems. The specific objectives of this thesis work can be stated as follows:

(1). For the proposed methodology of optimal shape design, it is required to develop and test a robust numerical scheme in order to solve Serret-Frenet equations. Serret-Frenet equations play an important role in the curve synthesis. Proper care must be taken in selecting the free and

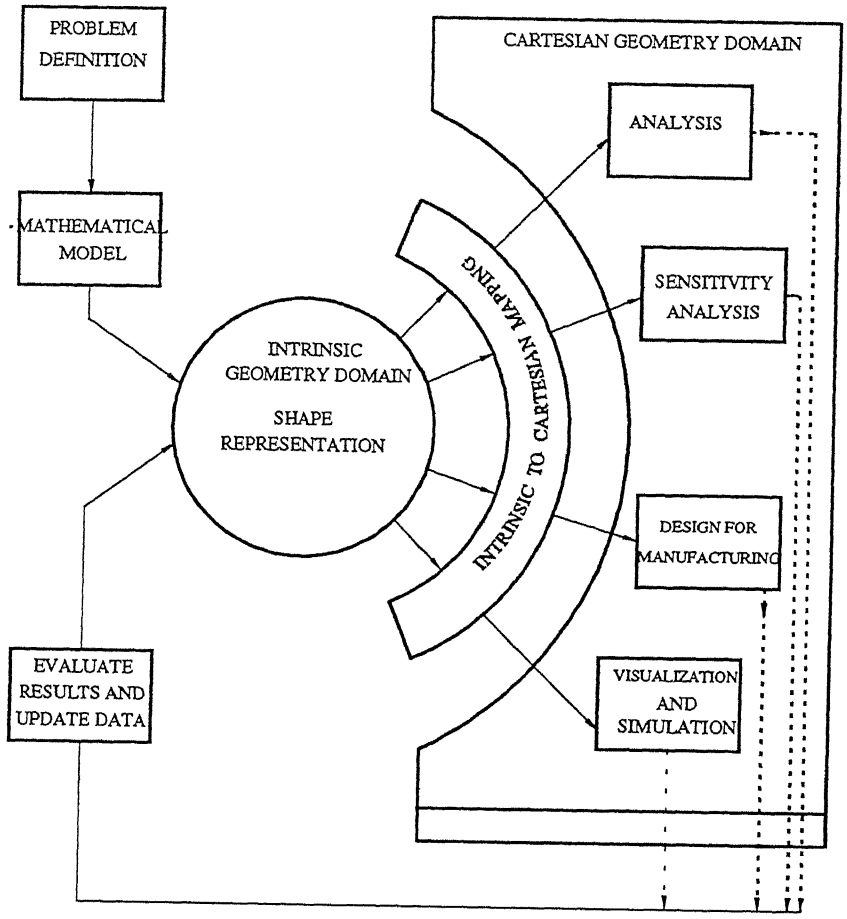


Figure 1.4: Role of shape in a typical engineering design cycle

dependent design variables. The robust numerical scheme eliminates the designer's intervention during the design process. Numerical integration schemes like Simpson's 1/3 rule and Gaussian quadrature rules will provide the necessary mapping from intrinsic to Cartesian domain by solving Serret-Frenet equations.

(2). A three-dimensional surface is developed as follows. First, two bounding curves are defined in space using LINCEs. These curves are selected such that they are frequently encountered in various engineering design problems. After the curve synthesis, a surface is generated by linearly

blending the above two curves in the *intrinsic domain* instead of the Cartesian domain. Since the surface is not defined entirely in terms of the intrinsic parameters like principal curvature, it is called a *pseudo-intrinsic* surface. Intrinsic to Cartesian mapping is done later by a suitable numerical integration scheme for further analysis.

(3). Presently, analysis of engineering problems requires either FEM/BEM approach or numerical analysis methods. All these techniques are carried out in the Cartesian space. The intrinsic geometry should provide the information about the Cartesian geometry and any change made through the intrinsic shape design variables should update the Cartesian geometry. A FEM code will be developed for obtaining the objective function value, e.g., deflection of a shell surface under a uniformly distributed loading.

(4). Optimization of the surface for some specific objectives will be carried out by the application of simple genetic algorithms. The objective function could be in terms of weight, stiffness of the above defined surface.

(5). The optimized surface will be realized using the rapid prototyping technique to have a feel of the above shape optimization methodology.

(6). Illustrative examples will be considered and solved in order to show the capabilities of the above shape optimal design cycle of *synthesis-optimization-realization.*

1.5 Literature Review

The need for optimal design is not a recent subject of interest in mechanics. Galileo, in his well-known *discorsi e dimostrazioni matematiche intorno a Due Nuove Scienze* (1638), presents the problem of finding the shape of a cantilever beam in order to obtain a uniform stress distribution for all the sections [43]. After this early attempt, different optimization problems were studied by Bernoulli, Euler, Lagrange, and Saint-Venant among many others. Many of the works in shape optimization are limited to two-dimensional problems only. Essentially, they have solved the problems of defining the outer boundary of a hole or a fillet etc. ([3], [12], [14], [15], and [45]).

Haftka and Grandhi [22], Ding [13] have reviewed the work of several researchers in the field of structural shape optimization in their review articles. Boundary shape has been represented in three ways: using boundary nodes; using polynomials; using spline or spline blending functions. Numerical solution methods used are: sequential linear programming; penalty function methods; feasible direction methods; optimality criteria; pattern transformation methods; and sequence of approximate problems.

Ding [12] has also studied the two-dimensional shape optimization of elastic structures with optimal thickness for fixed parts, using B-splines as shape functions and by adapting design element technique simultaneously. Ebrahimi [14] has used a continuously differentiable thickness function for the flywheel design. Shape optimal design using Bezier and B-spline curves is investigated by Shyy *et all* [37]. High-order elements for designing a hole in a plate under biaxial state of stress and a fillet are studied in their work, by using the finite element method.

The major drawback of parametric representation of a curve or surface (e.g. splines) is that the information about the curvature and/or torsion is not available explicitly. Not only that it results in too many design variables which leads to high cost and difficult optimization problems to solve. In contrast, review of current literature shows that intrinsic representation definitely involves lesser number of shape design variables than the parametric approach. In the literature on differential geometry [40], Cornu spiral, known as clothoid, but also called as transition spiral or a railway curve, since it is often used in rail track layout to increase or decrease the curvature linearly is probably the widely illustrated example curve for intrinsic geometry. Intrinsic definition of a Cornu spiral is a linear curve in the curvature-arc length plane.

The concepts of curve synthesis using linear curvature elements, termed as LINCEs, has been developed by Nutbourne [28]. This concept has been successfully implemented by Adams, Pal, Tavakkoli, Venkatesh, and Srinivasa Rao ([1], [29], [41], [44], and [39]). Venkatesh has solved 2-D shape optimization problems of structural and mechanical elements using BEM technique and zero-order and first order methods. A torque arm subjected to axial and transverse loading, a fillet under an axial load, and a ladle hook subjected to tensile loading were optimized in his research work. Tavakkoli and Dhande [42] have applied the intrinsic curve synthesis

technique to Variable Geometry Truss design problems. Srinivasa Rao has outlined the shape design of a three dimensional curve using two planar curves defined using clothoid(s). His method has been applied for designing the 2-D or 3-D path of a mobile robot or the trajectory of a manipulator arm.

Cheu *et all* [5] have studied the optimal design of gas turbine blades for minimum weight using a combined finite element and sequential linear programming technique. The blade was modelled (in 3-D) by using the quadratic shell elements of NASTRAN package. Geometric constraints were imposed on the thickness variation such that optimal design has smooth aerodynamic shape. Thus the profile of the blade is fixed, but the thicknesses of various elements are the shape design variables.

Very limited literature is available in three dimensional shape optimization. The only work found in the shape optimization of 3-D structural components is that of Imam [25]. He has studied the cantilever beam problem in detail by the FEM analysis and nonlinear mathematical programming techniques. He has proposed and implemented various techniques for shape representation like independent node movement, design element, super curves defined by polynomial expressions, and superposition of shapes.

As for as the application of GAs for optimization problems is concerned, a welded beam design problem consisting of a highly non-linear objective function viz. fabrication cost with five non-linear functional constraints was studied by Deb [10]. In his technical note, GAs are compared with some of the traditional optimization methods like successive linear approximation, geometric programming, simplex and random search methods. It is found that GAs have surprising speed of convergence to near-optimal solutions.

Optimization of composite laminate stacking sequence for carrying maximum buckling load was studied by Riche and Haftka [35]. The design space was found to include multiple optima, especially in case of large number of plies. GAs were applied to the structural optimization problems by Chaturvedi *et all* [4]. A real coded Genetic Algorithm has been used to tackle the continuous variables and penalty function method is used to handle the design constraints. For

size optimization, ten bar and eighteen bar trusses were considered and the results were found to be better than some of the classical methods. For topology or configuration optimization, a eighteen bar truss and a forty seven bar planar tower were considered. The results were found to be matching with the existing results. Several suggestions were given on the application of real GAs to more complex structural optimization problems. Minimum weight design of a trussed rafter structure and geometrical optimization of a 18 bar cantilever truss problems were studied by Jenkins [26]. Several other structural optimization problems were also studied in the same paper.

Rapid prototyping allows the designers to verify their work and developers to construct functional models quickly ([30] and [31]). A sub-optimal or optimal design can be easily verified at a reduced cost and time. The optimized surface shape will be realized using the FDM 1650 machine to get a feel of the design.

1.6 Organization

The organization of the thesis can be outlined as follows. Chapter 1 introduces the relevant background information about the present work. Various concepts of surface design using CAD have been reviewed. A brief review of optimization using Genetic Algorithms has been given. Application of surface design in various fields and review of literature relevant to the present work has been discussed. Scope and objectives of the present work, and the organization of the thesis have been drawn out.

Chapter 2 deals with pseudo-intrinsic shape design, by first introducing the concepts of intrinsic geometry applied to curve design, and then extending it to the proposed surface design. The design algorithm for the surface has been described along with a flow-chart and some examples were discussed at the end to illustrate the proposed shape synthesis approach.

Chapter 3 discusses the principle of genetic algorithms (GAs) and their application to the pseudo-intrinsic surface design. Computational issues involved in the use of GAs were outlined.

Application problems to illustrate the proposed shape design methodology have been taken up in Chapter 4. A minimum area surface and maximum stiffness surface have been discussed. The methodology adopted for the realization of the pseudo-intrinsic surface using the FDM-1650 rapid prototyping machine has been explained.

Chapter 5 deals with the results of the application of the proposed shape optimal design methodology for some interesting engineering applications. These include optimal design of the surface for minimum area (or weight) and for maximum stiffness. A brief summary of the work that has been done in this thesis and several suggestions for further work in this direction have been mentioned.

PSEUDO-INTRINSIC SHAPE DESIGN

2.1 Intrinsic Curve Design

This section introduces the basic concepts of curve design using the intrinsic geometry. Curve (or any geometric element) can be represented in four ways viz. parametric, implicit, explicit, and intrinsic form [36]. In the intrinsic form, a space curve in general requires two Equations $\kappa = \kappa(s)$ and $\tau = \tau(s)$ for its complete and unique description, where κ is the curvature, τ is the torsion and s is the arc length [28]. As opposed to the intrinsic form, neither of the above forms have the ability to control the intrinsic properties of the curve explicitly. If the intrinsic properties of a curve can be associated with some useful variables to be controlled, then the intrinsic shape synthesis definitely becomes more attractive.

At any point on a space curve there is an infinity of normal vectors, but only one tangent vector. The tangent vector is the derivative of the position vector $\mathbf{t}(s) = \mathbf{r}'(s)$ and has unit length. The derivative of the tangent vector defines both the principal normal vector $\mathbf{n}(s)$ and the curvature $\kappa(s)$. The vector cross product of tangent vector and normal vector defines the binormal vector $\mathbf{b}(s)$. The tangent vector, the normal vector, and the binormal vector are three mutually perpendicular unit vectors and define a curve frame $\mathbf{f}(s)$ that moves along the curve as the arc length increases. The torsion $\tau(s)$ is a measure of the magnitude and sense of deviation of a curve from the osculating plane in the neighbourhood of the corresponding point of the curve, or simply the rate of change of the osculating plane.

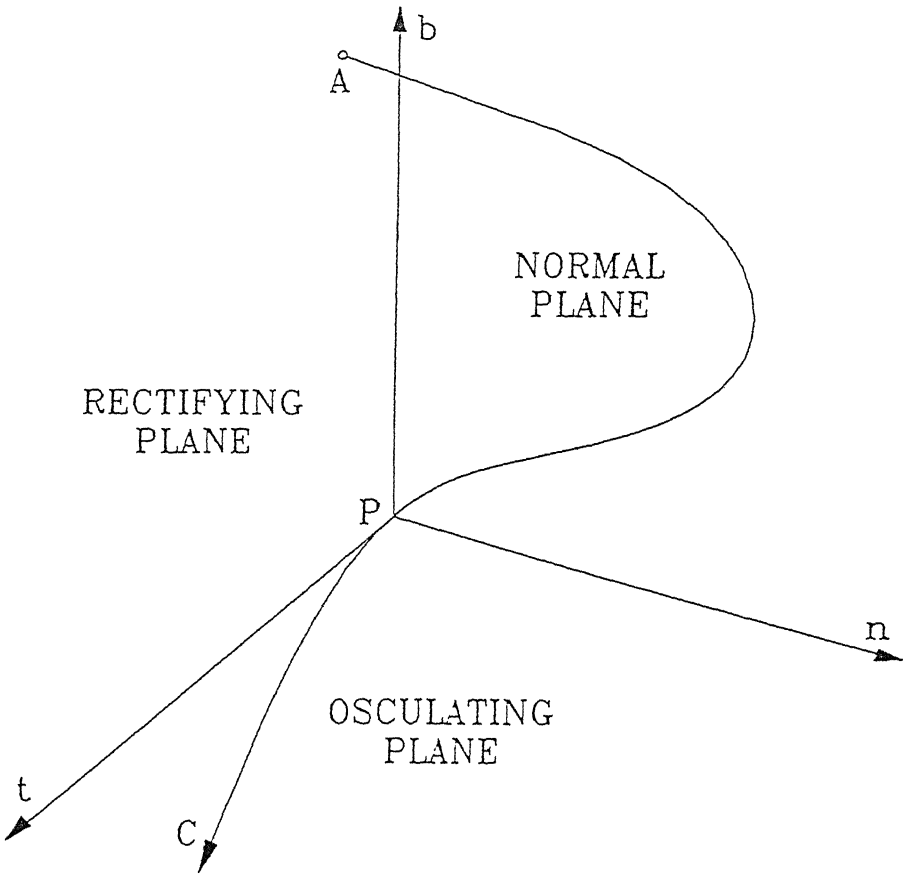


Figure 2.1: Local planes of moving trihedron of a Curve [45]

The plane containing **t** and **n** is called the osculating plane. The plane containing **b** and **n** is called the normal plane. And the plane containing **b** and **t** is the rectifying plane. These planes are shown in Figure 2.1.

The above definitions can be mathematically written as follows. Consider a curve **C** as shown in Figure 2.2. Let $\mathbf{r}(s)$ be the position vector of a generic point P and let s the arc length of P from a reference point A, be the parameter describing the curve as $\mathbf{r}(s)$. Then

$$\mathbf{t} = \frac{d\mathbf{r}}{ds} \tag{2.1}$$

$$\frac{d\mathbf{t}}{ds} = \kappa\mathbf{n} \tag{2.2}$$

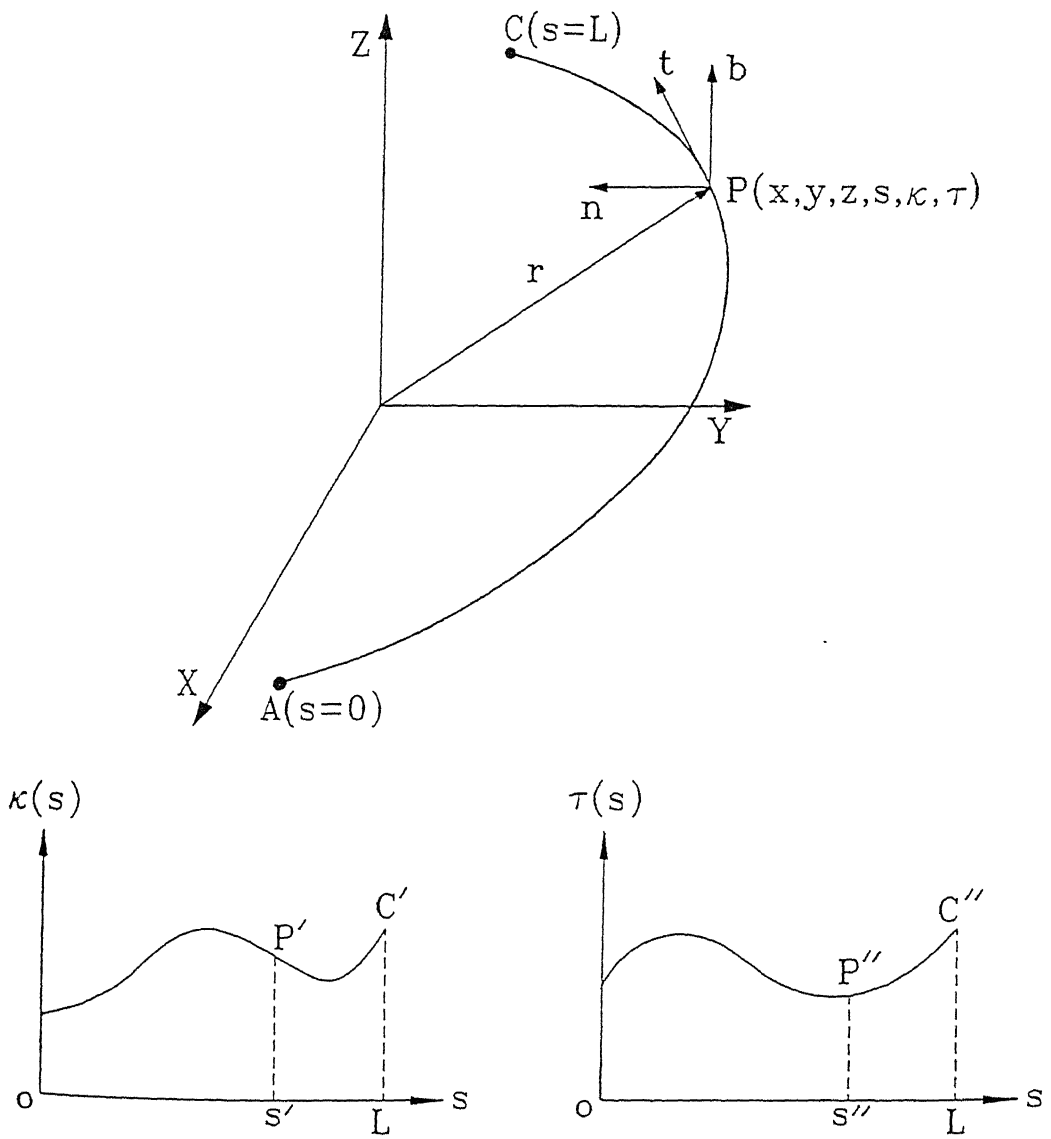


Figure 2.2: Intrinsic and Cartesian Geometry of a 3-D Curve [45]

$$\frac{d\mathbf{n}}{ds} = -\kappa\mathbf{t} + \tau\mathbf{b} \tag{2.3}$$

$$\frac{d\mathbf{b}}{ds} = -\tau\mathbf{n} \tag{2.4}$$

$$\mathbf{b} = \mathbf{t}\mathrm{x}\mathbf{n} \tag{2.5}$$

where **t**, **n**, and **b** are the unit tangent, normal, and binormal vectors respectively, κ is the curvature, and τ is the torsion.

Equations (2.2) to (2.4) are known as Serret-Frenet formulae. These Equations are used as the basis for the proposed shape synthesis in this thesis work.

If a curve is planar, then $\tau = 0$ and the above Equations can be written as follows:

$$\mathbf{r} = \begin{Bmatrix} x(s) \\ y(s) \end{Bmatrix} \tag{2.6}$$

$$\mathbf{t} = \frac{d\mathbf{r}}{ds} \quad \text{and} \quad \mathbf{n}\mathrm{x}\mathbf{t} = 0 \tag{2.7}$$

$$\frac{d\mathbf{t}}{ds} = \kappa\mathbf{n} \tag{2.8}$$

$$\frac{d\mathbf{n}}{ds} = -\kappa\mathbf{t} \tag{2.9}$$

The review of literature shows that different mathematical strategies have been adopted to find the general solution of the above Equations. The approach used in the present work is described in the following.

The problem of finding the coordinates (x, y) as a function of the arc length s can be solved by using the Serret-Frenet Equations (2.2) through (2.4) [26]. For 2-D curve synthesis, we start with Equation (2.7) as follows. Once again consider the curve **C** in Figure 2.2. Let us assume that the tangent angle at A has been specified as ψ_0 and the arc length of A as s_0. Let us rewrite Equation (2.7) as,

$$\mathbf{t} = \frac{d\mathbf{r}}{ds} = \left[\frac{dx(s)}{ds} \quad \frac{dy(s)}{ds}\right]^{\mathrm{T}} \tag{2.10}$$

Since $\mathbf{n}.\mathbf{t} = 0$, then

$$\mathbf{n} = \left[\frac{-dy(s)}{ds} \quad \frac{dx(s)}{ds}\right]^{\mathrm{T}} \tag{2.11}$$

on the other hand :

$$\frac{d\mathbf{t}}{ds} = \left[\frac{d^2x(s)}{ds^2} \quad \frac{d^2y(s)}{ds^2}\right]^{\mathrm{T}} \tag{2.12}$$

Substituting Equations (2.11) and (2.12) in Eq. (2.3) yields,

$$\left[\frac{d^2x(s)}{ds^2} \quad \frac{d^2y(s)}{ds^2}\right]^{\mathrm{T}} = \kappa(s)\left[\frac{-dy(s)}{ds} \quad \frac{dx(s)}{ds}\right]^{\mathrm{T}} \tag{2.13}$$

therefore,

$$\frac{d^2x(s)}{ds^2} + \kappa(s)\frac{dy(s)}{ds} = 0 \tag{2.14}$$

$$\frac{d^2y(s)}{ds^2} - \kappa(s)\frac{dx(s)}{ds} = 0 \tag{2.15}$$

where the boundary conditions at the starting point A are assumed to be known i.e., at $s = s_0$, $x = x_0$, $y = y_0$, and $\psi = \psi_0$.

Assume that $\kappa(s)$ is defined as a piecewise linear function of s. Let

$$\eta' = x' + iy' \tag{2.16}$$

where, $\eta' = \frac{d\eta}{ds}$, $x' = \frac{dx(s)}{ds}$, and $y' = \frac{dy(s)}{ds}$.

The governing Equations (2.14) and (2.15) can now be written as follows:

$$(\eta')' - i\kappa(s)(\eta') = 0 \tag{2.17}$$

$$\text{or} \qquad \eta' = e^{i\int \kappa(s)ds} = e^{i\psi(s)} \tag{2.18}$$

where $\int \kappa(s)ds = \psi(s)$.

Separating the real and imaginary parts, we get

$$\frac{dx(s)}{ds} = cos[\psi(s)] \tag{2.19}$$

$$\frac{dy(s)}{ds} = sin[\psi(s)] \tag{2.20}$$

Integrating these Equations with respect to the arc length s yields the parametric coordinates of the 2-D curve in terms of the intrinsic parameter s.

$$x(s) = \int_{s_0}^{s} \cos[\psi(\sigma)]d\sigma + x_0 \tag{2.21}$$

$$y(s) = \int_{s_0}^{s} \sin[\psi(\sigma)]d\sigma + y_0 \tag{2.22}$$

where

$$\psi(\sigma) = \int_{s_0}^{\sigma} \kappa(s)ds + \psi_0 \tag{2.23}$$

Equations (2.21) through (2.23) can be solved by using numerical integration and numerical nonlinear Equation solving schemes for the Cartesian coordinates $x(s)$ and $y(s)$ and the change in the tangent angle $\psi(s)$ along the 2-D curve once the curvature profile $\kappa(s)$ is known.

2.1.1 Planar Curve Design

We will now describe the planar curve design when the curvature profile is given as piecewise continuous linear segments i.e. LINCEs [28]. Let us consider the problem of defining a curve passing through two points $P_0(x_0, y_0)$ and $P_1(x_1, y_1)$ in a plane, and the tangent angles at P_0 and P_1 are specified as ψ_0 and ψ_1 respectively. This problem can be divided in to the following steps:

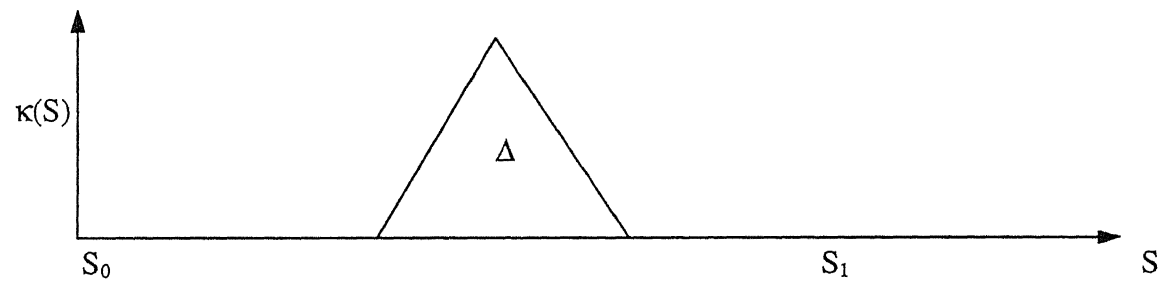
1. **Choosing A Shape Model**: The variation of the curvature κ as a function of the arc length i.e. *curvature profile* is defined. This step involves choosing a suitable shape model as it decides the geometry of the curve uniquely.

2 **Intrinsic to Cartesian Mapping**: Once the shape model is defined, Equations (2.21) through (2.23) are solved simultaneously for the complete definition of the curve in the Cartesian space.

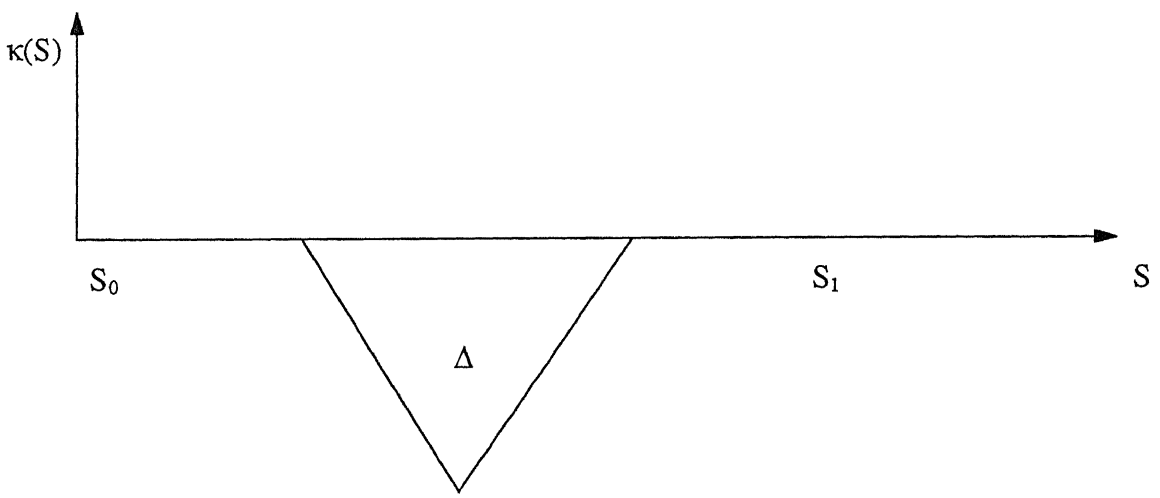
Shape Models:

Two types of shape models are chosen for the present study which will be described below [39].

Straight line Clothoid Straight line (S–C–S) model: In S–C–S model, two end portions of the curve are straight lines and they are blended with a clothoid pair (symmetric cornu spirals) in between. Two variations of this model are shown in Figure 2.3. Area of the triangle Δ represents the change in the tangent angle, therefore a triangle below the S-axis indicates a negative change in the tangent angle.

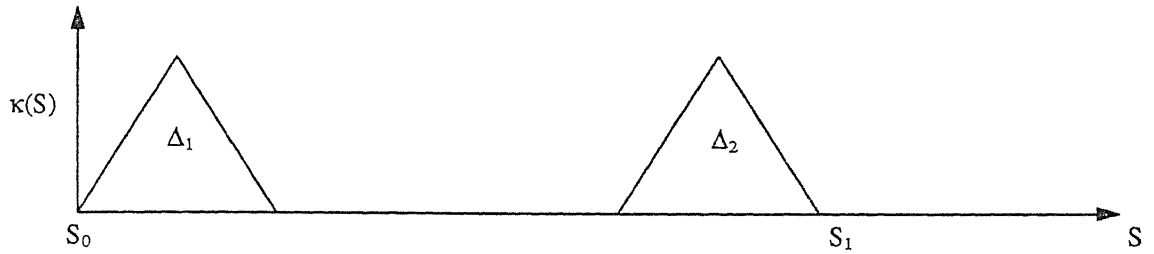


(a) Positive change in tangent angle

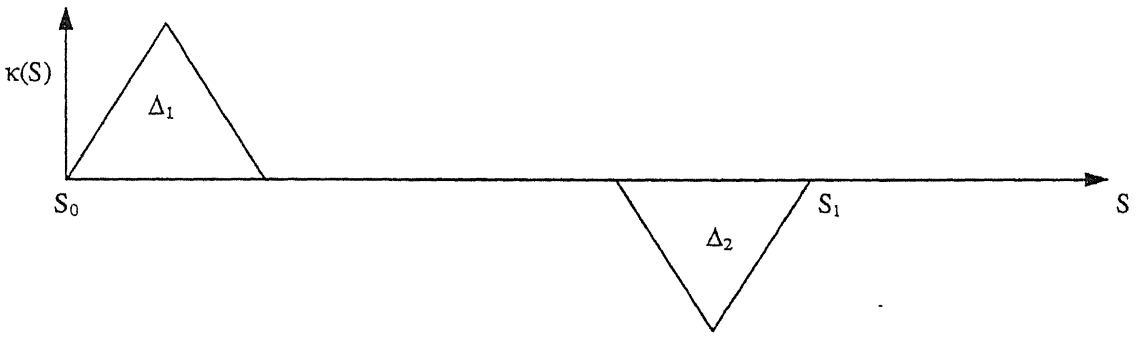


(b) Negative change in tangent angle

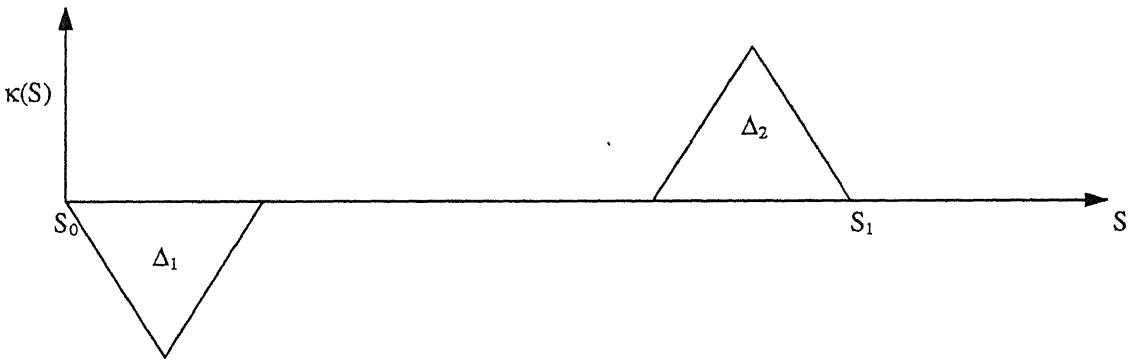
Figure 2.3: S–C–S Model with positive and negative changes in tangent angle



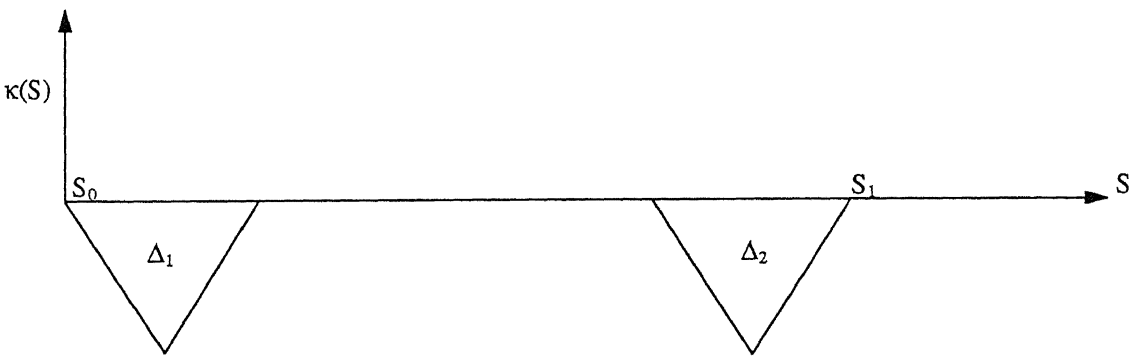
(a) Positive tangent angle change for both the clothoid pairs



(b) Positive tangent angle change for first the clothoid pair and negative for the second.



(c) Negative tangent angle change for first the clothoid pair and positive for the second.



(d) Negative tangent angle change for both the clothoid pairs

Figure 2.4: C–S–C shape model with possible combinations of positive and negative tangent angle changes

Clothoid Straight line Clothoid (C–S–C) model: In C–S–C model the two end portions of the curve are pairs of symmetric spirals and the middle portion is a straight line. There are four different 5 LINCEs models of this nature that are possible, as shown in Figure 2.4. Again the areas of the triangles Δ_1 and Δ_2 represent the change in the tangent angle of the first and second clothoid pairs respectively. The slope of the straight line portion is called as the sharpness λ.

Design Methodology for S–C–S Model:

The input design parameters are the two end points P_0 and P_1, the tangent angles ψ_0 and ψ_1, and the sharpness λ. First step is to find the intersection point $Q(x_4, y_4)$ of the two tangents as follows (Figure 2.6).

$$x_4 = \frac{(m_0 x_0 - m_1 x_1) + (y_1 - y_0)}{(m_0 - m_1)} \tag{2.24}$$

and

$$y_4 = m_0 (x_4 - x_0) + y_0 \tag{2.25}$$

where, $m_0 = \tan \psi_0$ and $m_1 = \tan \psi_1$ are the slopes of the initial and final tangents.

The variation of the sharpness λ over the arc length of the clothoid pair is shown in Figure 2.5, below.

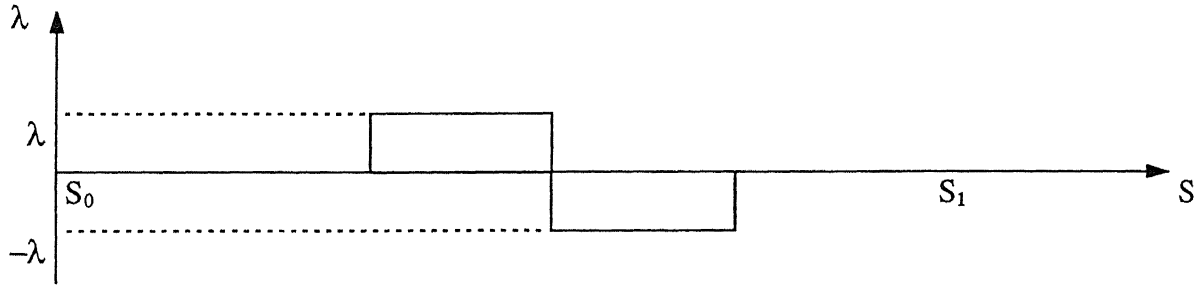


Figure 2.5: Variation of Sharpness Vs Arc length for S–C–S model of Figure 2.3(a)

Let s' be the total arc length of the clothoid pair which blends the two tangents and $\alpha = \psi_1 - \psi_0$ be the area under the κ–s curve, then from the geometry of Figure 2.3(a),

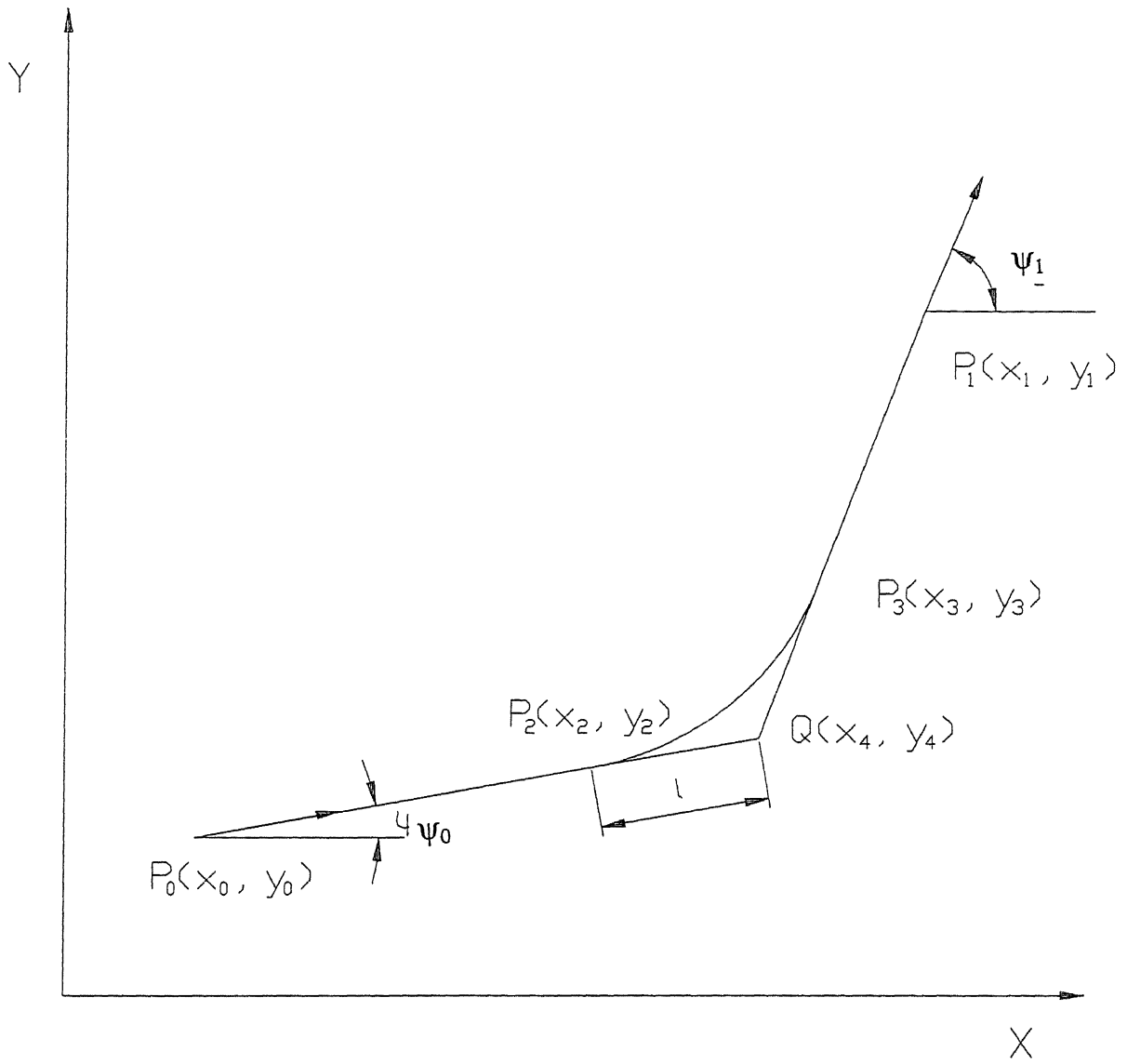


Figure 2.6: Cartesian geometry of S–C–S Shape model of Figure 2.3(a).

$$\alpha = \frac{\lambda s'^2}{4} \tag{2.26}$$

or

$$s' = 2\sqrt{\frac{\psi_1 - \psi_0}{\lambda}} \tag{2.27}$$

Since $\kappa(s)$ is linear function of s, we substitute $\kappa_0 + \lambda s$ for $\kappa(s)$ in Equations (2.21) through (2.23). By rotating the axes so that the X-axis is coincident with the initial tangent and noting the fact that the curve starts with zero curvature, we get the following expressions for the Cartesian coordinates of the clothoid pair:

$$x_3 - x_2 = \int_0^{s'} \cos(\lambda t^2 / 2)dt \tag{2.28}$$

$$y_3 - y_2 = \int_0^{s'} \sin(\lambda t^2 / 2)dt \tag{2.29}$$

From the geometry (Figure 2.6) $l(\cos\psi_0 + \cos\psi_1) = x_3 - x_2$, therefore

$$l = \frac{(x_3 - x_2)}{(\cos\psi_0 + \cos\psi_1)} \tag{2.30}$$

where l is the exit distance of the clothoid pair.

Now P(x_2, y_2) can be calculated from the following expressions:

$$\left.\begin{aligned} x_2 &= x_4 - l\cos\psi_0 \\ y_2 &= y_4 - l\sin\psi_0 \end{aligned}\right\} \tag{2.31}$$

By calculating all the unknowns, the points on the curve at regular intervals of the parameter arc length s can be calculated.

Design Methodology for C–S–C Model:

The input parameters are the same as before. Let ψ_m be the angle made by the intermediate straight line portion of the curve as shown in Figure 2.7. It is assumed that the sharpness of both the clothoid pairs at the ends is same and they are symmetric within themselves. From the same

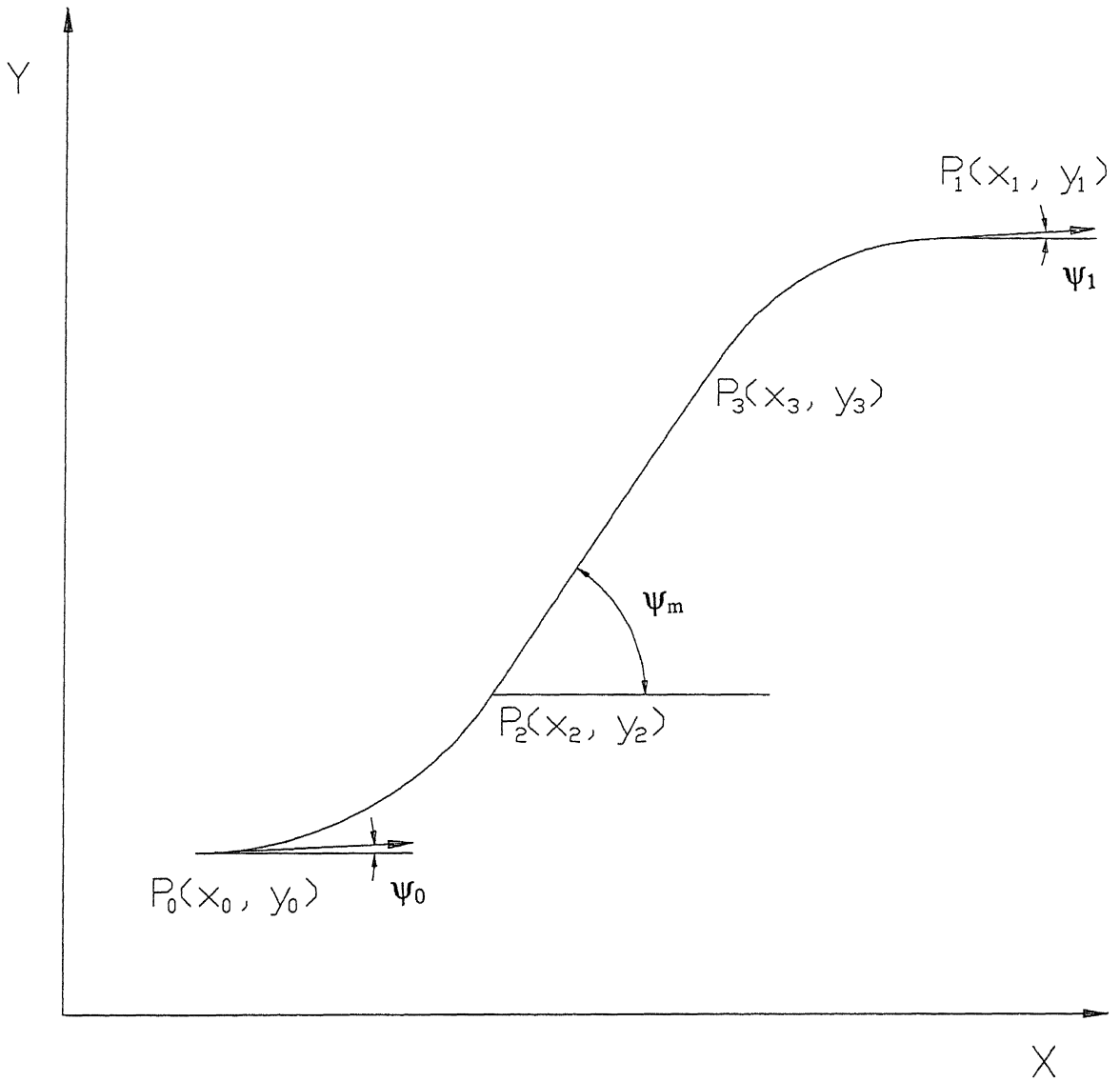


Figure 2.7: Cartesian geometry of C–S–C Shape model of Figure 2.4(b).

arguments given for the S–C–S model, we can write for the first clothoid pair:

$$\psi_m - \psi_0 = \frac{\lambda s_2^{\,2}}{4} \quad \text{or} \quad s_2 = 2\sqrt{\frac{\psi_m - \psi_0}{\lambda}} \tag{2.32}$$

Similarly for the second clothoid pair also,

$$\psi_1 - \psi_m = \frac{\lambda (s_1 - s_3)^2}{4} \quad \text{or} \quad s_3 = s_1 - 2\sqrt{\frac{\psi_1 - \psi_m}{\lambda}} \tag{2.33}$$

To compute the values of s_2 and s_3 it is necessary to know the value of ψ_m before hand. This is computed iteratively as follows.

STEP 1: Since there is no basis to guess the value of ψ_m, the initial angle is taken equal to the angle made by the line joining the initial and final points.

STEP 2: With this value of ψ_m, s_2 and s_3 are computed. Subsequently $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$ are also computed.

STEP 3: ψ_m will be updated to the value equal to the line joining $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$.

STEP 4: Compare the old value of ψ_m with the updated value. If the difference is less than some acceptable accuracy, this new value is finalized and is used for all further calculations. Otherwise go to step 2.

It has been found that the above method results in faster convergence than taking the initial value of ψ_m equal to a constant value. Once the value of ψ_m is calculated, the points on the curve can be calculated by using the expressions (2.21) through (2.23). Simpson's 1/3 rule has been used to carry out the numerical integration of these Equations.

2.2 Pseudo-intrinsic Surface Design

Surfaces can also be represented in four ways similar to the curves viz. parametric, implicit, explicit, and intrinsic form. A surface can be generated parametrically by the movement of the tip of a position vector [28],

$$\mathbf{r} = \mathbf{r}(u, v) \tag{2.34}$$

where u, v are independent parameters each varying smoothly over a specified range of values usually 0 to 1.

Non-parametric surfaces can be either in explicit or in implicit form. An implicit Equation of a surface is given by F(x, y, z) = 0; similarly, a surface is explicitly represented as z = f(x, y) just by solving the implicit Equation for one of the variables as a function of the other two. This non-parametric representation has the inability to represent an easily transformable and bounded surface. This weakness is easily overcome by the parametric representation. But the parametric representation of surfaces, similar to curves, lacks the explicit control over the intrinsic parameters like normal and principal curvatures etc. If these intrinsic quantities can be related to some design variables, this will become a great advantage in the field of 3-D shape optimization due to reduced number of decision variables.

A surface can be defined in terms of the intrinsic quantities like normal and principal curvatures, and tangent plane. An intrinsic surface is uniquely determined by its **first** and **second fundamental forms**. Nutbourne [28] has given a detailed description of intrinsic surfaces in his book. However, we confine ourselves only to the approach followed in the present work.

The surface generated in this thesis work is similar to a ruled surface. A ruled surface can be generated either by a family of curves or by joining corresponding points on two space curves. The expression for the latter approach, which is used in this study, is given by [16]

$$\mathbf{r} = (1 - \xi)\,\mathbf{r}_1(u) + \xi\,\mathbf{r}_2(u) \tag{2.35}$$

where $\mathbf{r} = \mathbf{r}_1(u)$ and $\mathbf{r} = \mathbf{r}_2(u)$ are the two space curves and ξ is a parameter in the range (0, 1).

Here the two curves are blended linearly in the *Cartesian domain*. However, in the present work, the surface is generated by linearly blending the two curves that are defined in terms of the intrinsic parameters viz. curvature and arc length in the *intrinsic domain*. Thus the surface is not completely defined using the intrinsic geometry and hence it is called a *pseudo-intrinsic* surface.

There can be different combinations of these two bounding curves from the shape models that are described in the preceding section, but with different parameter values. For example, (i) S-C-S and S-C-S, (ii) C-S-C and C-S-C, (iii) S-C-S and C-S-C and so on. The reason behind choosing these two curves is because of the fact that these curves are frequently encountered in many of the engineering design problems. The first two are relatively simple when compared to the third one. Hence, we want to illustrate the proposed surface design methodology by considering the S-C-S and C-S-C case. Refer Figures 2.6 and 2.7. Let $\kappa_1(s)$ and $\kappa_2(s)$ are the curvature functions of the first and second bounding curves, respectively. Let $\kappa(s)$ be the curvature function of a curve at a distance ξ from the first curve, then

$$\kappa(s) = (1 - \xi)\, \kappa_1(s) + \xi\, \kappa_2(s) \tag{2.36}$$

where, $0 \leq \xi \leq 1$

Now, the curvature function $\kappa(s)$ of enough number of curves in between the two bounding curves can be calculated at regular intervals of the parameter ξ by using the above Equation so that a smooth surface is generated. Once the curvature function of a curve is known, the intrinsic to Cartesian mapping can be done using Equations (2.21) through (2.23).

2.3 Design Algorithm

Now we will describe the methodology of the proposed pseudo-intrinsic surface design. Let the C-S-C and S-C-S curves lie in two parallel XY planes lying at $Z=Z_1$ and $Z=Z_2$, respectively, as shown in Figure 2.8. Let us also assume that both the curves are designed between the same initial and final points i.e., X and Y coordinates of the initial point of the two curves are identical and also X and Y coordinates of the final point of the two curves are identical. Let $\kappa_1(s)$ and $\kappa_2(s)$ are the curvature functions of the S-C-S and C-S-C curves, corresponding to the parameter $\xi=0$ and $\xi=1$, respectively, as shown in Figure 2.9. Then, $\kappa(s)$ the curvature function of a curve at a distance ξ from the first curve is calculated by using Equation (2.36). Thus, by varying the value of ξ we can obtain all the interpolating curves.

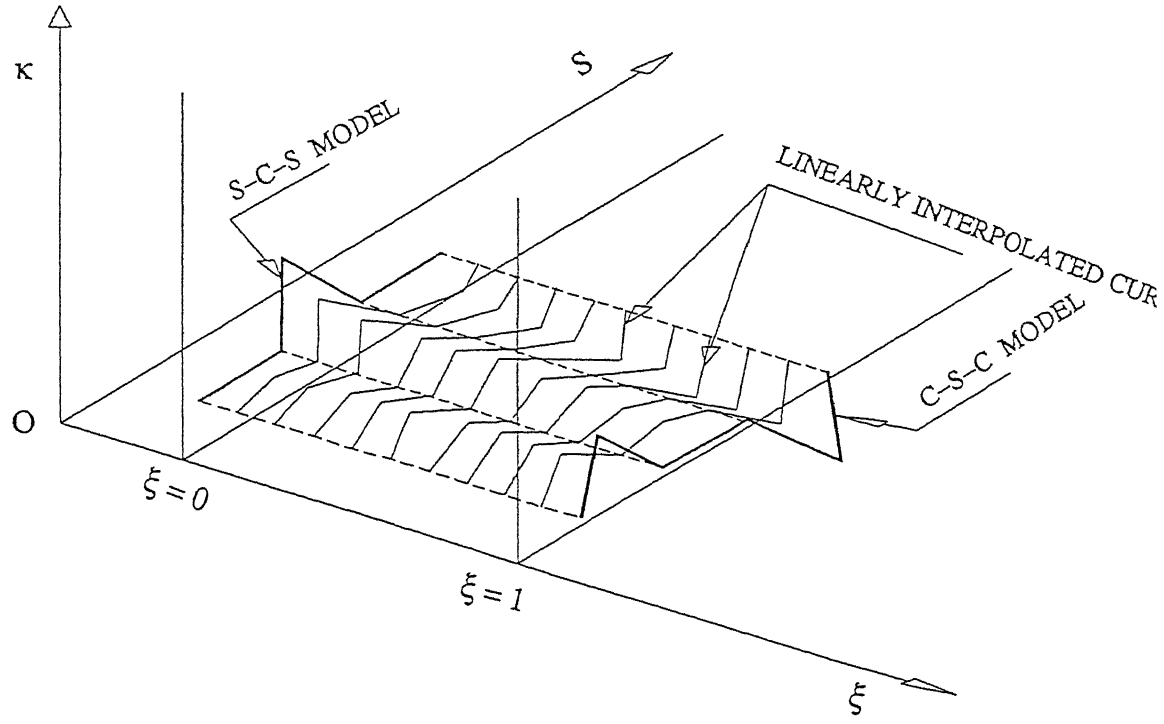


Figure 2.8: Curvature functions of S-C-S and C-S-C Shape models. These two are linearly interpolated along the ξ axis.

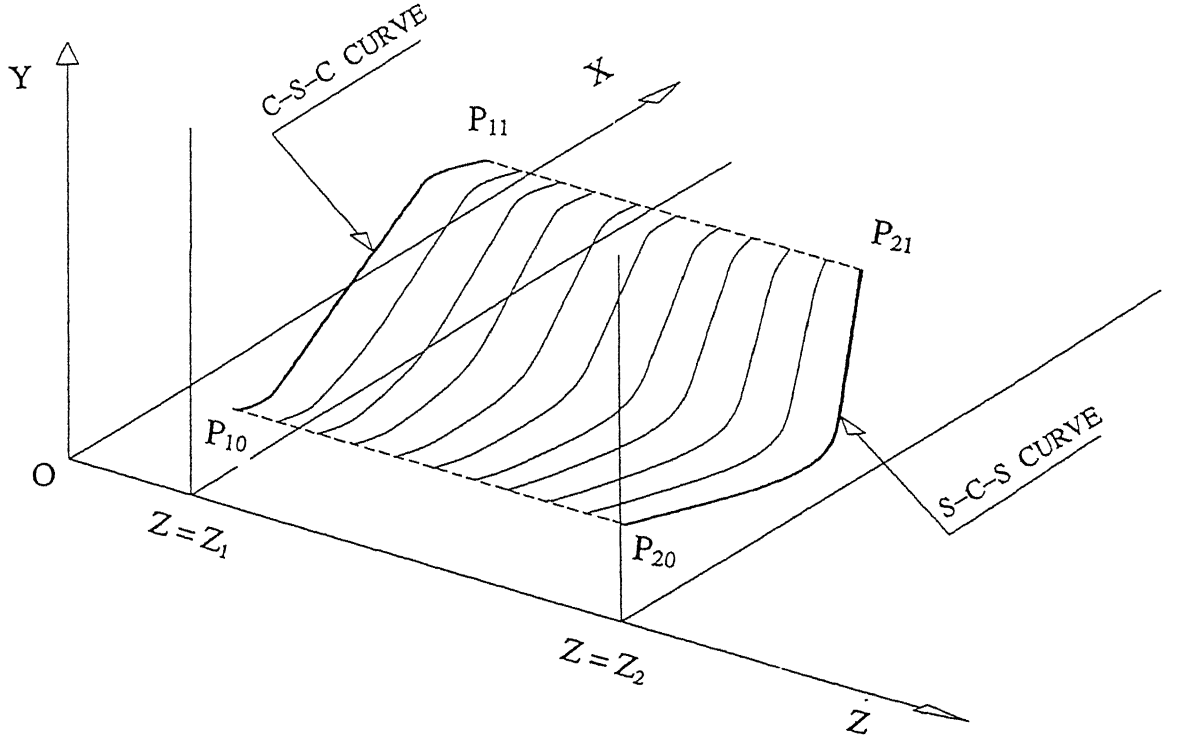


Figure 2.9: Cartesian geometry of S-C-S and C-S-C Shape models along with the resulting surface. Just for clarity S-C-S curve is drawn away from the origin.

Without losing the generality of the above procedure, we can assume that the two *bounding curves* lie on $Z_1=0$ and $Z_2=\overline{Z}$ planes and pass through $P_0(x_0, y_0)$ and $P_1(x_1, y_1)$ points. Flow-chart for the design of the pseudo-intrinsic surface is given in Figure 2.10. Then the design variables are the sharpness and the tangent angles for each of the two curves, and the initial and final end points. With these variables as input, each curve is synthesized and the curvature values at enough number of equidistant points along the arc length, say 100 are calculated. Then these two curves are linearly blended using Equation (2.36). Each linearly interpolated curve is mapped in to the Cartesian space by the numerical integration of Equations (2.21) through (2.23). Now the surface can be represented by an array of N_1x N_2 points lying on it, where N_2 is the number of points or rulings along each curve and N_1 is the number of curves in between and including the two bounding curves.

2.4 Some Examples

First we will consider the examples for curve synthesis. Figure 2.11 shows curves designed using the S-C-S shape model (Figure 2.3(a)) with sharpness as a parameter. All three curves pass through the same initial and final points and have the same tangent vectors at both the ends. The initial and final points are (0, 0) and (80, 60), respectively. The slope of the tangent vector at the beginning is 5^0 and at the end it is 60^0. Three cases i.e. λ = 0.001, 0.01, 0.1 are shown. It can be observed that by decreasing the sharpness of the clothoid pair we can achieve a smooth blending between the two straight line portions. As sharpness decreases the curvature decreases or the radius of curvature of the clothoid curve increases.

Similarly, Figure 2.12 shows the effect of the sharpness parameter λ on a C-S-C model curve. All the curves are again designed with same initial and final points and with same tangent vectors at the ends. They are designed with (0, 0) and (60, 60) as the initial and final points, respectively. The tangent vectors considered are 10^0 at both the two ends. Three cases i.e. λ = 0.005, 0.01, 0.1 are shown. The effect of λ is also similar to the earlier case.

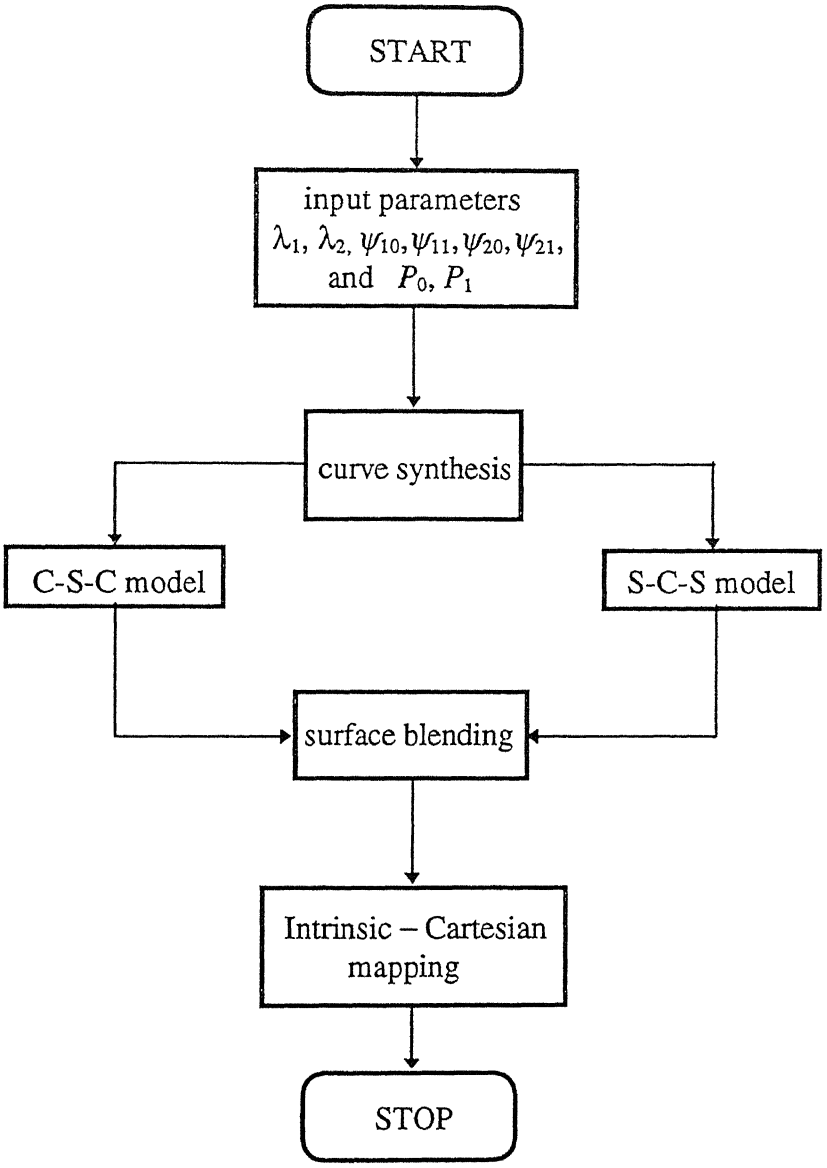


Figure 2.10: Flow-chart of the Pseudo-intrinsic Surface Design

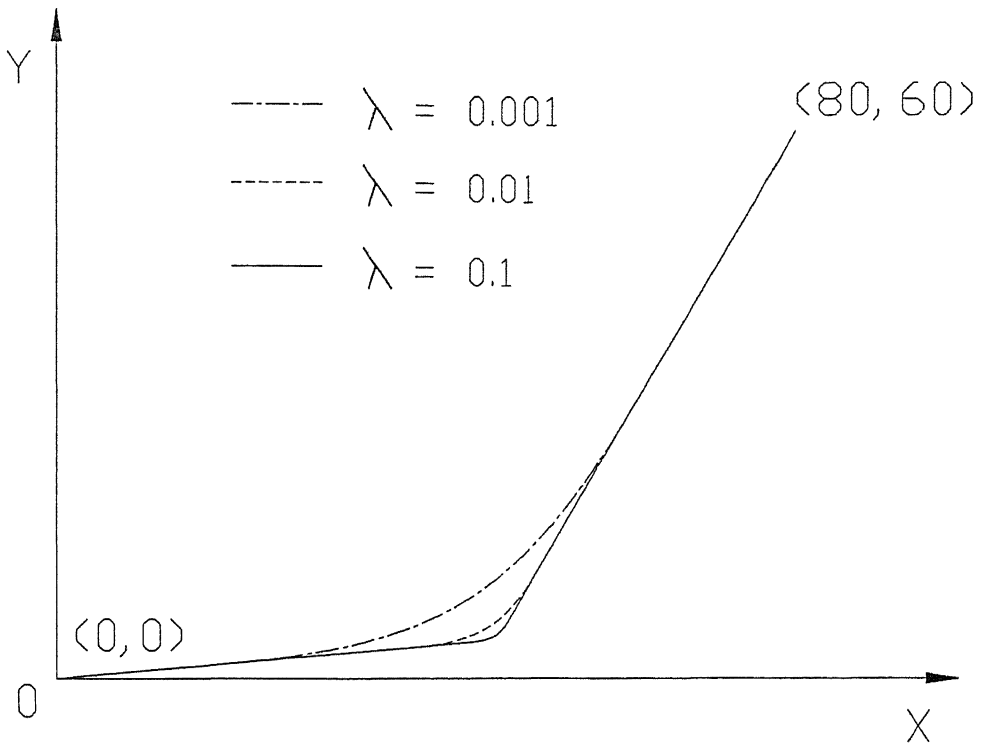


Figure 2.11: S-C-S Model with different values of λ, but with same end points and tangent vectors.

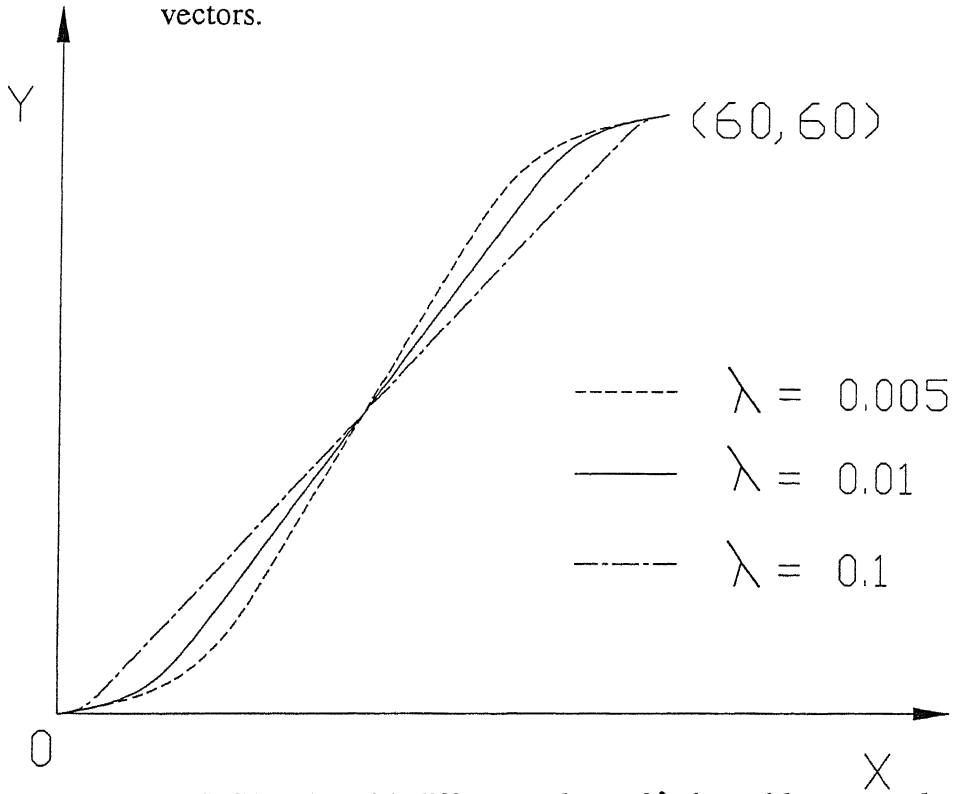


Figure 2.12: C-S-C Model with different values of λ, but with same end points and tangent vectors.

Finally, Figure 2.13 shows a pseudo-intrinsic surface designed using $\lambda_1 = 0.001$, $\lambda_2 = 0.01$, $\psi_{10} = 10^0$, $\psi_{11} = 60^0$, $\psi_{20} = 5^0$, $\psi_{21} = 5^0$ as the design variables. The two bounding curves pass through the same initial and final points of (0, 0) and (100, 100), respectively. A total of 11 curves along C_1 parametric direction and 11 rulings along the C_2 direction are considered in this example. If we use more number of curves and rulings we can get a surface with the curvatures varying from point to point smoothly. Also by controlling the above design variables we can design a surface to suit a particular application. This topic will be discussed in the next chapter.

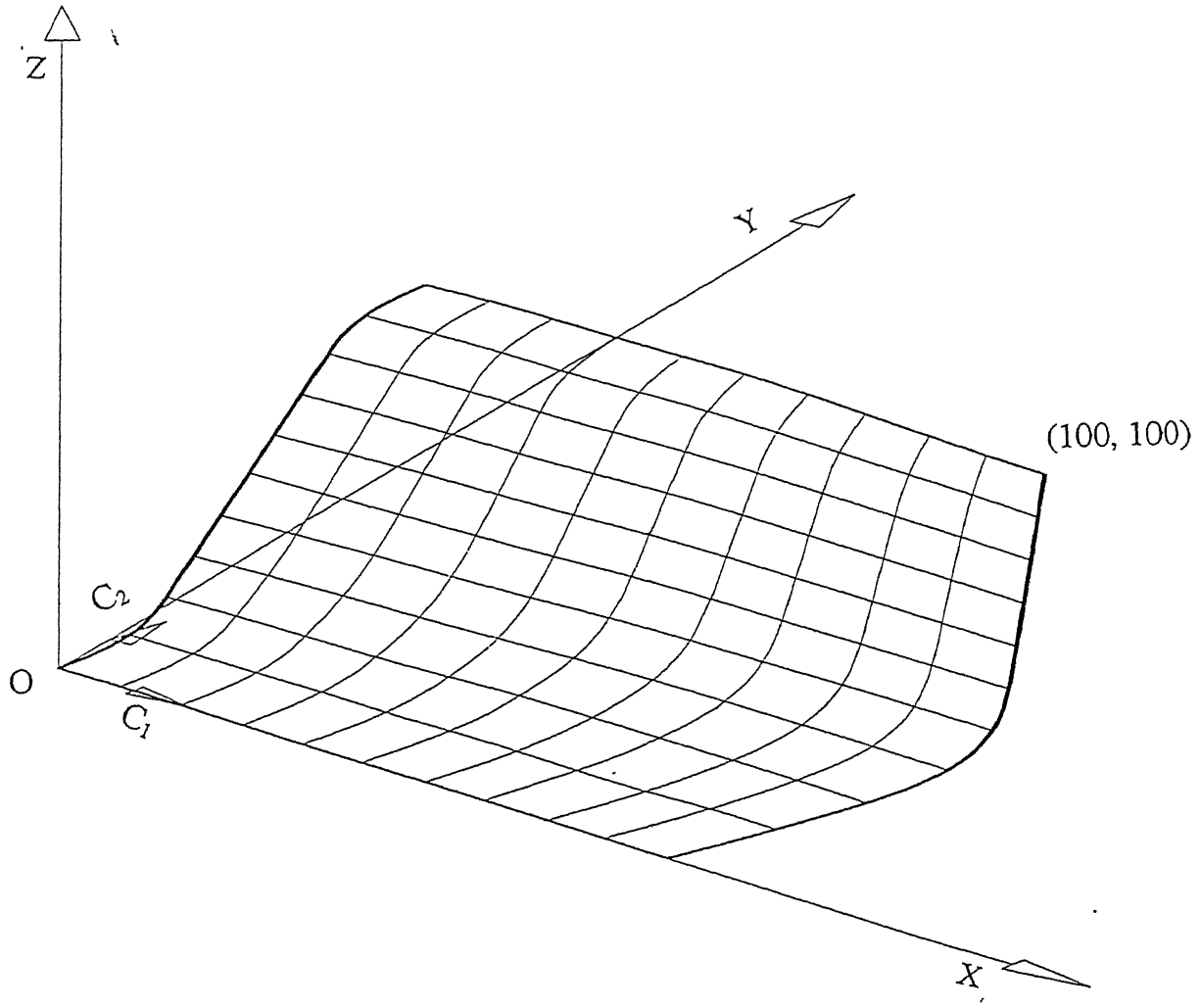


Figure 2.13: Pseudo-intrinsic Surface with $\lambda_1 = 0.001$ and $\lambda_2 = 0.01$.

2.5 Closure

In this chapter, we have started with the basic concepts of intrinsic geometry. Synthesis of curves using the intrinsic parameters like curvature and torsion is discussed. The methodology of designing planar curves relevant to the present thesis work has been covered along with the shape models that are chosen. Pseudo-intrinsic surface design is described and the design algorithm is presented. Simple example problems are given to depict the use of the proposed shape synthesis. In the next chapter, we will describe the principle of genetic algorithms (GAs) and their application to the shape optimization problems using the methodology developed in this chapter.

SHAPE OPTIMIZATION USING GENETIC ALGORITHMS

3.1 Introduction to Genetic Algorithms

A Genetic Algorithm (GA) can be understood as an "intelligent" probabilistic search algorithm which can be applied to a variety of combinatorial optimization problems [6]. The theoretical foundations of GAs were originally developed by Holland [23]. GAs are based on the evolutionary process of biological organisms in nature. During the course of evolution, natural populations evolve according to the principles of natural selection and "survival of the fittest". Individuals which are more successful in adapting to their environment will have better chance of surviving and reproducing, whilst individuals which are less fit will be eliminated. This means that the genes from the highly fit individuals will spread to an increasing number of individuals in each successive generation. The combination of good characteristics from highly adapted ancestors may produce even more fit offspring. In this way, species evolve to become more and more well adapted to their environment. Before looking at the basic steps involved in genetic algorithms and how they work with a variety of problems in engineering, sciences and commerce, it is better to review the working and some of the pitfalls of the so called traditional or classical optimization methods.

Consider a one dimensional minimization problem. Most of the traditional optimization algorithms begin with an initial point (feasible or infeasible depending on the algorithm) and then improve the solution through a series of successive iterations [11]. Usually a search direction is used from the starting point to find a point that corresponds to a minimum function value in that

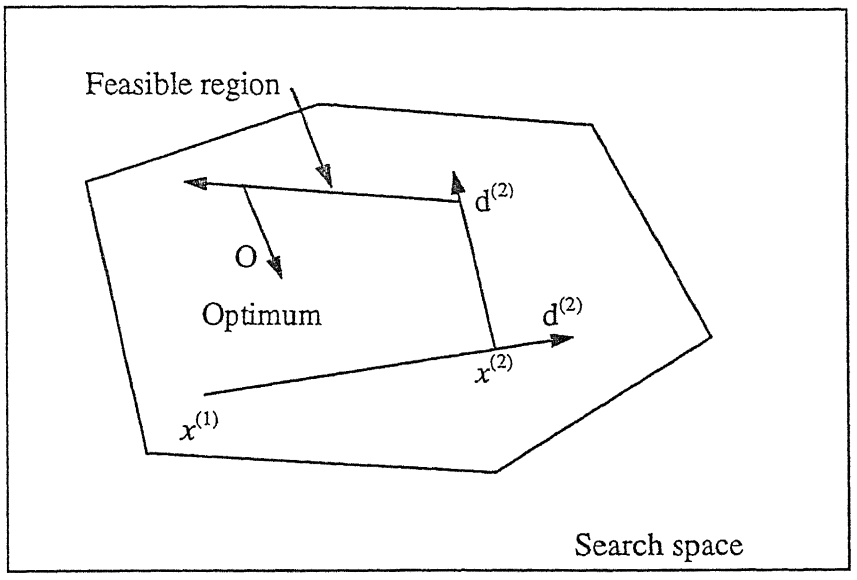


Figure 3.1: A typical optimization process. The initial point is $x^{(1)}$ and the initial search direction is $d^{(1)}$. The search process continues by finding an optimal point ($x^{(2)}$) along $d^{(1)}$ and then searching along another direction $d^{(2)}$ from $x^{(2)}$ [11]

direction by using a one dimensional search technique. If the new point found is $x^{(1)}$, a new search direction is found at this point and another one dimensional search is performed to locate the best point in that direction. This process is continued for a number of iterations until a set of predefined convergence criteria are satisfied. This procedure is illustrated in Figure 3.1. Most optimization methods vary in the way the search direction is chosen.

Even though there exist a number of algorithms, they can be broadly classified into two groups - Direct search methods and gradient based methods. The direct search methods use only the objective function values and do not depend on any gradient information and are therefore more flexible and can be used to a broad class of problems. Gradient based methods require the derivative information of the objective function and/or constraints either exactly or numerically. Among the direct search methods, Powell's method uses conjugate directions as search directions whereas Hooke and Jeeves method uses a history of previous points to create a search direction adaptively. Random search methods are simple in principle and in general require more function evaluations to obtain a desired accuracy than non-random methods. Among the gradient-based methods, Fletcher and Reeves method uses conjugate gradients as search directions whereas

Davidon-Fletcher-Powell (DFP) method uses Newton's method by adaptively approximating the underlying Hessian matrix. There are plenty of other optimization methods which can be found in any of the optimization books ([2], [11], [33], and [44]). Some algorithms are faster than the others because of the way the search direction is created. The choice of this algorithm over that primarily depends on experience and availability of software packages.

Even though these classical optimization algorithms have been extensively used in many engineering design and decision-making problems, there are a number of shortcomings. Most of the popular methods require the gradient information, which may not be easy to calculate (or not at all available) in many real-world problems. Moreover, most of these methods may converge to a locally optimal solution which may be very different from the actual global optimal solution. The most difficult problems among all is probably that none of these methods can be applied to a wide variety of problems. In order to alleviate some of these problems, a couple of non-traditional search and optimization algorithms - genetic algorithms and simulated annealing - are finding wide spread applicability in engineering design and decision making problems. These methods work according to the principles of natural phenomenon and found to have solved many nonlinear programming (NLP) problems to global optimality successfully [18]. In the next section, we describe the working principles of genetic algorithms.

3.2 Genetic Algorithms - Working Principle

As described in the preceding section, Genetic Algorithms are search and optimization procedures that are motivated by the principles of natural genetics and natural selection. Some fundamental ideas of genetics are borrowed and used artificially to construct search algorithms that are robust and require minimal problem information. Here, we describe the working principle of GAs. Consider a typical unconstrained single variable optimization problem which can be stated as [9]:

$$\text{Maximize:} \quad f(x) \tag{3.1}$$

$$\text{Variable bounds:} \quad x_{min} \leq x \leq x_{max}$$

In order to use GAs to solve the above problem, the variable x is typically encoded into a string or *chromosome* structure which represents a possible solution to the given problem. Binary coding is mostly used and the length of the string is usually determined by the accuracy of the solution desired. For example, if five bit binary strings are used to code the variable x then the string (0 0 0 0 0) is decoded to the value x_{min}, the string (1 1 1 1 1) is decoded to the value x_{max}, and any other string is decoded to a value in the range (x_{min}, x_{max}) uniquely. Thus, with five bit strings used to code the variable x, there will be a total of 2^5 or 32 different strings that are possible, and the accuracy between two consecutive strings will be (x_{max} - x_{min})/32. If more accuracy is desired longer strings can be used. With a known coding, any string can be decoded to a x value, which can be used to find the objective function value. A string's objective function value (f(x)) is called as the string's *fitness*. A pseudo code for a simple genetic algorithm is given in Figure 3.2. GAs begin with a population of strings (individuals) created at random [6]. Fitness of each individual is evaluated with respect to the given objective function. Then this initial population is operated by three main operators - *reproduction*, *crossover*, and *mutation* - to create hopefully a better population. Highly fit individuals or *solutions* are given opportunities to reproduce by exchanging pieces of their genetic information, in a crossover procedure, with other highly fit individuals. This produces a new "offspring" solutions (i.e. *children*), which share some characteristics taken from both the parents. Mutation is often applied after crossover by altering some *genes* (i.e. bits) in the strings. The offspring can either replace the whole population (*generational* approach) or replace less fit individuals (*steady-state* approach). This new population is further evaluated and tested for some termination criteria. If the termination criteria are not met, this reproduction-crossover-mutation-evaluation cycle is repeated until the termination criteria are met. One cycle of these three operators and evaluation procedure is called a *generation* in GA terminology. Before discussing the application of genetic algorithms to the present work, let us briefly discuss these three GA operators for a better understanding of the GAs.

Reproduction selects the good strings in a population and forms a mating pool for the subsequent crossover. There exist a number of reproduction operators in GA literature, but the essential idea is that above average individuals are picked from the current population and

```
begin
        Initialize population;
        Evaluate population;
        repeat
                Reproduction;
                Crossover;
                Mutation;
                Evaluate population;
        until (termination criteria);
end.
```

Figure 3.2: A Pseudo-code for a Simple Genetic Algorithm (SGA) [6].

duplicates of them are inserted in the mating pool. The commonly used reproduction operators are proportionate selection with roulette-wheel and tournament selection. In proportionate selection, a string is selected with a probability proportional to its fitness. For a fixed population size N, the probability of selecting an i-th individual with f_i fitness is $\frac{f_i}{Nf_{avg}}$, where f_{avg} is the average fitness of all the strings in the current population. One way to achieve this proportionate selection is to use a roulette-wheel with the circumference marked for each string proportionate to the string's fitness. The roulette-wheel is spun N times, each time keeping an instance of the string selected by the roulette-wheel pointer in the mating pool. This roulette-wheel mechanism is expected to insert f_i/f_{avg} copies of the i-th string. In tournament selection, some number of individuals (usually two) are randomly chosen from the population (with or without replacement). The best individual from this set is selected for further genetic processing, and this is repeated until the mating pool is filled.

Crossover operator is applied next to the strings in the mating pool. Here, two strings are picked at random from the mating pool and some portion of the information are exchanged between the strings. For example, in a single point crossover operator, this is performed by randomly choosing a crossing site along the string and by exchanging all bits on the right side of the cross site as shown below:

$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{array} \Rightarrow \begin{array}{cccccc} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{array}$$

It is intuitive from this construction that good substrings from either parent string can be combined to form a better child string if an appropriate site is chosen. Since, the knowledge of an appropriate site is usually not known, a random site is chosen. With a random site the children produced may or may not have combination of good substrings from parent string depending on whether the cross site fall in appropriate place or not. But we do not worry about this aspect very much, because if good strings are created by crossover, there will be more copies of them in the next mating pool generated by reproduction operator. But if good strings are not created by crossover they will not survive beyond next generation, because reproduction will not select those strings. In a two point crossover operator, two random sites are chosen and contents bracketed by these sites are exchanged between two parents. There is another operator known as uniform crossover in which each bit from either parents is selected with a probability of 0.5. It is worthwhile to mention that the purpose of the crossover operator is two-fold. First, it searches the parameter space. The other aspect is that the search needs to be performed in such a way that the information stored in the parents is maximally preserved, because these parents are the instances of good strings selected by reproduction operator. In the single point crossover, the search is not extensive but maximum information is preserved from parent to children. On the other hand, in uniform crossover the search is very extensive but minimum information is preserved between parents and children. In order to preserve some good strings found in the mating pool, not all strings in the population are used in crossover. If a crossover probability of p_c is used, then $(p_c \times 100)\%$ of strings are used for crossover and the rest of the strings are simply copied to the new population.

Crossover operator is mainly responsible for the *search aspect* in GAs, even though Mutation operator is also used sparingly. Mutation changes a 1 to a 0 and vice-versa with a small mutation probability p_m. The need for mutation is to keep diversity in the population. For example, if in a particular position along the string length all strings in the population have a value 0, and a 1 is needed in that position to obtain the optimum then neither reproduction nor

crossover will be able to create this. But mutation introduces some probability of turning that 0 into a 1. Furthermore, mutation may be found useful for local improvement of a solution.

3.3 Optimization of Pseudo-Intrinsic Surface - Methodology

The problem of shape optimization can be described as minimizing or maximizing an objective function subject to a certain set of constraints. The objective function is defined in terms of some geometric design variables. In order to limit the search space constraints in the form of lower and upper bounds on the design variables are imposed. In the proposed work, the geometric design variables or Shape Design Variables (SDVs) are the following - sharpness of the clothoid pair(s), tangent angles at both the ends of each curve. It is assumed that both these curves are designed between the same initial and final points. Let $P_0(x_0, y_0)$ and $P_1(x_1, y_1)$ be the initial and final points, λ_1 be the sharpness of the clothoid pair of the S-C-S model and λ_2 be the sharpness of both clothoid pairs of the C-S-C model, ψ_{10} and ψ_{11} be the slopes of the tangent vector at the ends for the S-C-S model, and ψ_{20} and ψ_{21} be the corresponding values for the C-S-C model. Then the shape design vector can be written as,

$$\mathbf{X} = \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \psi_{10} \\ \psi_{11} \\ \psi_{20} \\ \psi_{21} \end{Bmatrix} \tag{3.2}$$

In order to use GA, the unconstrained optimization problem is stated as follows:

$$\text{Maximize:} \quad F(\mathbf{X}) \tag{3.3}$$

$$\text{Subject to:} \quad X_i^l \le X_i \le X_i^u \tag{3.4}$$

where X_i^l and X_i^u are the lower and upper bounds on variable X_i . X_i is the i-th component of the vector $\mathbf{X}$ given by equation (3.2).

Fitness of the above problem for a given design vector $\mathbf{X}$ is the objective function value at $\mathbf{X}$ itself. Even though the above problem is of maximization, a minimization problem can be solved by taking the fitness as $\frac{1}{1+F(\mathbf{X})}$.

Since, in GAs the initial population of design vectors or designs is generated randomly, a particular combination of the design variables may not be a possible solution. For example, let us try to design a S-C-S curve passing through (0, 0) and (30,40). Let us suppose the initial tangent angle is 0^0 and the final tangent angle is 45^0. Now, using equation (2.24) and (2.25) we obtain the intersection point of the two straight line portions as (-10, 0) which is just not a desired curve. i.e. physically not possible or feasible. In this case the end tangent slope can not be less than 60^0 to have a possible blending between the two end lines. Hence such combination of designs have to be discarded in the subsequent generations. This is done by setting the fitness of that particular individual (design) to reasonably a large value in case of minimization problems and to a very small value in case of maximization problems. This ensures the selection of only the feasible designs for performing the GA operations.

A Simple Genetic Algorithm (SGA) code that was developed by Goldberg [19] is used in the present study. The steps involved in the optimization (using GA) of the pseudo-intrinsic surface as designed using the methodology developed in chapter 2 can be stated as follows:

STEP 1: Initialize the GA parameters. This involves specifying the population size, maximum number of generations, string length of each variable, crossover and mutation probabilities etc. Upper and lower limits on each of the shape design variable should also be specified. Initialize the generation number to 0.

STEP 2: Generate an initial feasible random population.

STEP 3: For each individual in the population, synthesize the pseudo-intrinsic surface as described in Section 2.4. Define intrinsic to Cartesian mapping and find the Cartesian coordinates using equation (2.21) through (2.23).

STEP 4: Once the Cartesian coordinates are available, it is possible to analyze for the required optimality criterion e g., surface area or stiffness etc This may require use of some FEM code also Evaluate the fitness of each individual Infeasible designs have to be discarded at this stage as explained earlier.

STEP 5: Once the fitnesses of all the individuals is available, GA operations can be performed. This includes doing reproduction, crossover, and mutation one after the other. Now a new set of designs is created which possibly is better than that of the previous generation. This completes one generation

STEP 6: Increment the generation number. If the current generation number is greater than the maximum number of generations **terminate** Otherwise **go to step 3**.

3.4 Computational Issues

In general, considerable amount of time is consumed by the analysis routine in many of the shape optimization problems. The analysis routine may contain a FEM/BEM simulation before obtaining the required objective function. In genetic algorithms the computational time basically depends on the population size and the number of generations for which the GA code is being run. If N is the population size, G is the maximum number of generations, and p_c is the crossover probability then the total time of GA run would be roughly $N \times G \times p_c$.

3.5 Closure

In this chapter, we have started with an overview of the traditional optimization methods and then the working principle of GAs has been discussed. Application of a Simple Genetic Algorithm to the proposed shape design has been described. Computational efforts that are required are also given. In the next chapter, we will discuss the evaluation of the objective function for the pseudo-intrinsic surface considering with respect to some engineering application.

APPLICATIONS OF THE PROPOSED METHODOLOGY

4.1 Minimum Area Surface

The surface can be thought of as an assembly of quadrilateral patches. The surface area of each patch is calculated considering it as a trapezium and added to get the total surface area of the surface. Assuming a constant thickness throughout the surface, this objective is also equivalent to minimum weight. Every designer starts with such a simple objective to test his methodology before applying to a more complex optimality criterion.

4.2 Maximum Stiffness Surface

Deflection of a body is a measure of the strain energy absorbed within it. Designing a structure for maximum stiffness may be interpreted as meaning the determination of that structure of given volume, which stores, for given loads, the least amount of strain energy [24]. Using this concept, we have minimized the deflection of the surface as shell surface subjected to uniform pressure loading. The shell surface is assumed as an assembly of small, flat rectangular elements as shown in Figure 4.1. As the subdivision decreases convergence occurs [47]. The finite element formulation of such a surface is discussed below.

Consider a typical polygon flat element (Figure 4.2) subjected to *in-plane* and *bending* actions simultaneously, in local coordinates. The state of strain in in-plane action is completely

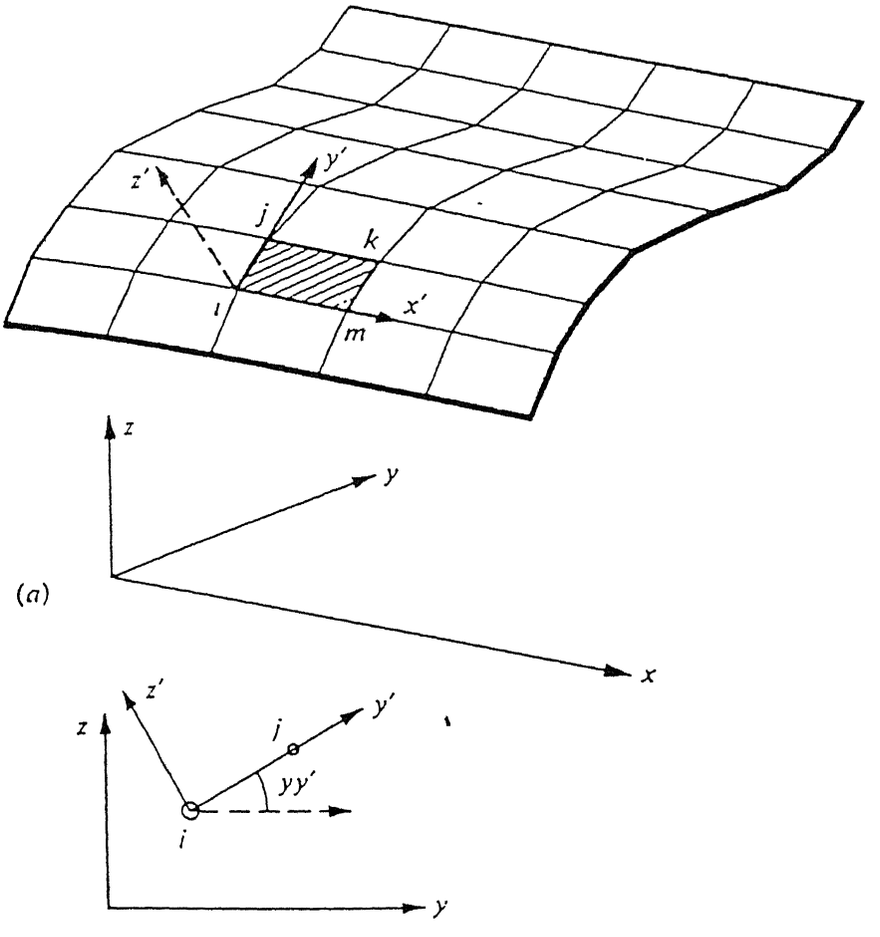


Figure 4.1: A shell surface as an assembly of rectangular elements. Local and global coordinates [47]

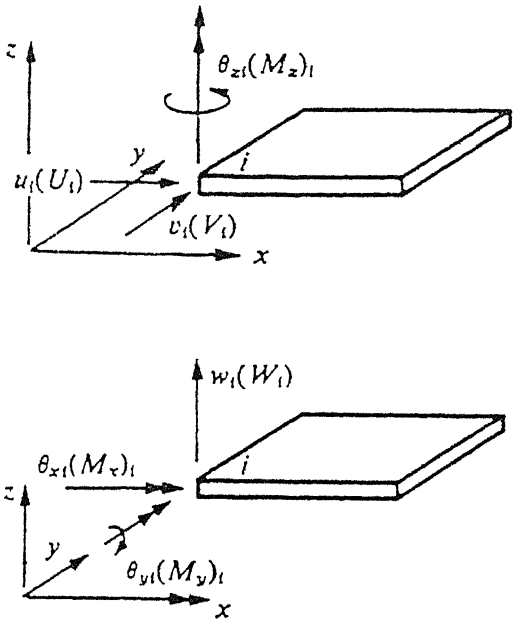


Figure 4.2: A typical rectangular flat element in local coordinates subjected to 'in-plane' and 'bending actions' simultaneously [47]

described in terms of u and v displacements of each typical node *i*. The minimization of the total potential energy leads to the following Equation.

$$\mathbf{f}^{ep} = \mathbf{K}^{ep}\mathbf{a}^{p} \quad \text{with} \quad \mathbf{a}_i^{p} = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \quad \text{and} \quad \mathbf{f}_i^{p} = \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} \tag{4.1}$$

where U_i and V_i are the nodal in-plane forces and $\mathbf{K}^{ep}$ is the in-plane stiffness matrix given by

$$\mathbf{K}^{ep} = \iint_{v^e} \mathbf{B}^{T}\mathbf{D}\mathbf{B}dxdy \tag{4.2}$$

where **B** is the strain-displacement matrix and **D** is the material matrix which will be given later.

Similarly, if **N** is the shape function matrix and q is the intensity of the uniformly distributed pressure load, then $\mathbf{f}_i^{p}$ can be directly obtained from the following expression.

$$\mathbf{f}_i^{p} = -\iint_{v} \mathbf{N}^{T}qdxdy \tag{4.3}$$

Similarly, for bending action, the state of strain was given by the nodal displacement in the z-direction w and the two rotations θ_x and θ_y. This results in

$$\mathbf{f}^{eb} = \mathbf{K}^{eb}\mathbf{a}^{b} \quad \text{with} \quad \mathbf{a}_i^{b} = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} \quad \text{and} \quad \mathbf{f}_i^{b} = \begin{Bmatrix} W_i \\ M_{xi} \\ M_{yi} \end{Bmatrix} \tag{4.4}$$

where W_i, M_{xi} and M_{yi} are the nodal force, moment about x-axis, and moment about y-axis, respectively, and $\mathbf{K}^{eb}$ is the bending stiffness matrix given by Equation (4.2) except for **B** and **D** have to be for bending case.

These two Equations are combined by introducing rotation θ_z associated with a fictitious couple M_z, for the sake of convenience, as follows:

Redefining the combined nodal displacements as

$$\mathbf{a}_i = \begin{Bmatrix} u_i \\ v_i \\ w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{zi} \end{Bmatrix} \tag{4.5}$$

and the appropriate 'forces' as

$$\mathbf{f}_i^{e} = \begin{Bmatrix} U_i \\ V_i \\ W_i \\ M_{xi} \\ M_{yi} \\ M_{zi} \end{Bmatrix} \tag{4.6}$$

Then we can write

$$\mathbf{f}^{e} = \mathbf{K}^{e}\mathbf{a}^{e} \tag{4.7}$$

where $\mathbf{K}^{e}$ is the elemental stiffness matrix given by,

$$\mathbf{K}_{rs} = \begin{bmatrix} \mathbf{K}^{p}_{rs} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} & \mathbf{K}^{b}_{rs} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 & 0 \end{matrix} & 0 \end{bmatrix} \tag{4.8}$$

4.2.1 Transformation to the Global Coordinates

The stiffness matrix given by Equation (4.8) has to be transformed to the global coordinate system. Refer to Figure 4.3 for the local and global coordinate systems denoted by xyz and $x'y'z'$ respectively.

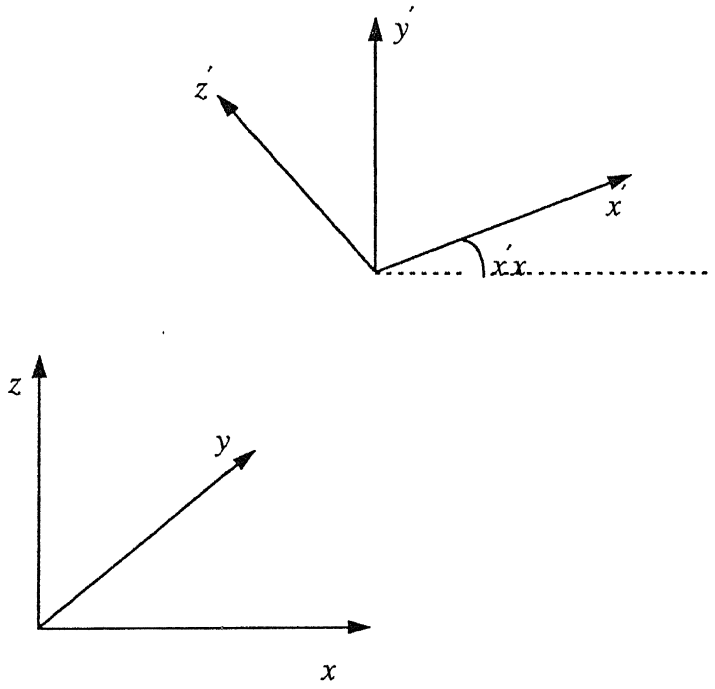


Figure 4.3: Local and global coordinates

The forces and displacements at a node transform from the global to the local system by a matrix **L** giving

$$\mathbf{a}'_i = \mathbf{L}\mathbf{a}_i \qquad \mathbf{f}'_i = \mathbf{L}\mathbf{f}_i \tag{4.9}$$

in which

$$\mathbf{L} = \begin{bmatrix} \lambda & \mathbf{0} \\ \mathbf{0} & \lambda \end{bmatrix} \tag{4.10}$$

with λ being the 3×3 matrix of direction cosines of angles formed between the two sets of axes i.e.,

$$\lambda = \begin{bmatrix} \lambda_{x'x} & \lambda_{x'y} & \lambda_{x'z} \\ \lambda_{y'x} & \lambda_{y'y} & \lambda_{y'z} \\ \lambda_{z'x} & \lambda_{z'y} & \lambda_{z'z} \end{bmatrix} \tag{4.11}$$

in which $\lambda_{x'x}$ = cosine of angle between the x' and x axes, etc.

For the whole set of forces acting on all the nodes of an element we can therefore write

$$\mathbf{a}^{\mathrm{fe}} = \mathbf{T}\mathbf{a}^{\mathrm{e}}, \text{ etc.} \tag{4.12}$$

By the rules of orthogonal transformation the stiffness matrix of an element in the global coordinates becomes

$$\mathbf{K}^{\mathrm{e}} = \mathbf{T}^{\mathrm{T}}\mathbf{K}'^{e}\mathbf{T} \tag{4.13}$$

In both of the above Equations **T** is given by

$$\mathbf{T} = \begin{bmatrix} \mathbf{L} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{L} & \mathbf{0} & \\ \mathbf{0} & \mathbf{0} & \mathbf{L} & \\ \vdots & & & \end{bmatrix} \tag{4.14}$$

a diagonal matrix built up of **L** matrices in a number equal to that of the nodes in the element.

Now the typical stiffness submatrix becomes

$$\mathbf{K}^{\mathrm{e}}_{\mathrm{rs}} = \mathbf{L}^{\mathrm{T}}\mathbf{K}'^{\mathrm{e}}_{\mathrm{rs}}\mathbf{L} \tag{4.15}$$

in which $\mathbf{K}'^{\mathrm{e}}_{\mathrm{rs}}$ is determined by Eq. (4.8) in the local coordinates.

Similarly, if the origins of the local and global systems is the same, then

$$\begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} = \lambda \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \tag{4.16}$$

4.2.2 Local Direction Cosines:

We consider the simple rectangular element with one side of it parallel to the global x axis as illustrated in Figure 4.1. For a typical element *ijkm*, the direction cosines of x' axis are

$$\begin{aligned} \lambda_{x'x} &= 1 \\ \lambda_{x'y} &= 0 \\ \lambda_{x'z} &= 0 \end{aligned} \tag{4.17}$$

the direction cosines of the y' axis have to be obtained by considering the coordinates of various nodal points. Thus

$$\begin{aligned} \lambda_{y'x} &= 0 \\ \lambda_{y'y} &= +\frac{y_J - y_I}{\sqrt{\left[(z_J - z_I)^2 + (y_J - y_I)^2\right]}} \\ \lambda_{y'z} &= +\frac{z_J - z_I}{\sqrt{\left[(z_J - z_I)^2 + (y_j - y_I)^2\right]}} \end{aligned} \tag{4.18}$$

Similarly for the same section, we have for the z' axis

$$\begin{aligned} \lambda_{z'x} &= 0 \\ \lambda_{z'y} &= -\frac{z_J - z_I}{\sqrt{\left[(z_J - z_I)^2 + (y_J - y_i)^2\right]}} \\ \lambda_{z'z} &= +\frac{y_J - y_I}{\sqrt{\left[(z_J - z_i)^2 + (y_J - y_i)^2\right]}} \end{aligned} \tag{4.19}$$

Clearly the numbering of nodes in a consistent fashion is important to preserve the correct signs of the expressions.

4.2.3 Derivation of Stiffness Matrix and Right Side Vector

Consider a typical 4-noded rectangular element shown in Figure 4.4 below, with its sides of 2a and 2b in length.

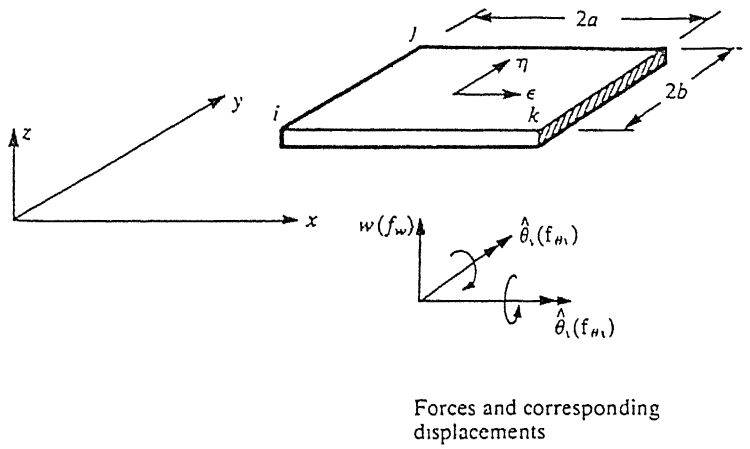


Figure 4.4: A typical rectangular element in local coordinates [47]

(i) in-plane or plane stress condition:

The approximation for the two displacements u and v at any point is given by ([7] and [20])

$$\begin{aligned} u &= a_0 + a_1 x + a_2 y + a_3 xy \\ v &= b_0 + b_1 x + b_2 y + b_3 xy \end{aligned} \tag{4.20}$$

in which a_0, b_0, ... are the constants to be found from the nodal coordinates of the element.

The shape functions in terms of the natural coordinates ξ and η are given by

$$N_i = \frac{(1+\xi\xi_i)(1+\eta\eta_i)}{4} \qquad \text{for } i = 1.,4 \tag{4.21}$$

where $\xi_i = \eta_i = -1,+1,+1,-1$, the values at the four corner nodes and ξ and η are the values at the gauss point under consideration.

The strain-displacement matrix is given by

$$[\mathbf{B}] = \left[\mathbf{J}^{-1}\right]^{\mathrm{T}} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & 0 & \frac{\partial N_2}{\partial \xi} & 0 & \frac{\partial N_3}{\partial \xi} & 0 & \frac{\partial N_4}{\partial \xi} & 0 \\ 0 & \frac{\partial N_1}{\partial \eta} & 0 & \frac{\partial N_2}{\partial \eta} & 0 & \frac{\partial N_3}{\partial \eta} & 0 & \frac{\partial N_4}{\partial \eta} \\ \frac{\partial N_1}{\partial \xi} & \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \xi} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \tag{4.22}$$

where $[\mathbf{J}]$ is the jacobian matrix given by

$$[\mathbf{J}] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{4.23}$$

in which

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi} x_i \tag{4.24}$$

and so on.

Now we can formulate the elemental stiffness matrix by using the gauss-legendre numerical integration by choosing appropriate number of gauss points in either direction.

$$[\mathbf{K}]^{ep} = \sum_i \sum_j w_i w_j \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} |\mathbf{J}| t \tag{4.25}$$

in which w_i and w_j are the weights associated with the i-th and j-th gauss points, respectively and t is the thickness of the element. The material matrix **D** for plane-stress condition is given by

$$[\mathbf{D}] = \frac{\mathrm{E}}{1-\upsilon^2} \begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & \frac{1-\upsilon}{2} \end{bmatrix} \tag{4.26}$$

where E is the modulus of elasticity and υ is the poisson's ratio.

(ii) Bending condition:

The approximation for the displacement w is given by [47]

$$w = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + a_6x^3 + a_7x^2y + a_8xy^2 + a_9y^3 + a_{10}x^3y + a_{11}xy^3 \tag{4.27}$$

Shape functions at the *i*-th node in natural coordinates are given by

$$\begin{aligned} N_i = \frac{1}{8}\{&(1+\xi_0)(1+\eta_0)(2+\xi_0+\eta_0-\xi^2-\eta^2), \\ &a\xi_i(1+\xi_0)^2(\xi_0-1)(1+\eta_0), \\ &b\eta_i(1+\xi_0)(\eta_0-1)(1+\eta_0)^2\} \end{aligned} \tag{4.28}$$

where

$$\begin{aligned} \xi_0 &= \xi\xi_i \\ \eta_0 &= \eta\eta_i \end{aligned} \tag{4.29}$$

The expression for the strain/displacement matrix can be obtained as

$$[\mathbf{B}] = \begin{bmatrix} \left(\frac{\partial\xi}{\partial x}\right)^2 & \left(\frac{\partial\eta}{\partial x}\right)^2 & \frac{\partial\xi}{\partial x}\frac{\partial\eta}{\partial x} \\ \left(\frac{\partial\xi}{\partial y}\right)^2 & \left(\frac{\partial\eta}{\partial y}\right)^2 & \frac{\partial\xi}{\partial y}\frac{\partial\eta}{\partial y} \\ 2\frac{\partial\xi}{\partial x}\frac{\partial\xi}{\partial y} & 2\frac{\partial\eta}{\partial x}\frac{\partial\eta}{\partial y} & \frac{\partial\xi}{\partial x}\frac{\partial\eta}{\partial y}+\frac{\partial\xi}{\partial y}\frac{\partial\eta}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial^2 Ni}{\partial\xi^2} \\ \frac{\partial^2 Ni}{\partial\eta^2} \\ 2\frac{\partial^2 Ni}{\partial\xi\partial\eta} \end{bmatrix} \tag{4.30}$$

in which

$$\frac{\partial x}{\partial\xi} = \sum_{i=1}^{12}\frac{\partial N_i}{\partial\xi}x_i \tag{4.31}$$

and so on.

Now the jacobian matrix can be calculated by using Equation (4.23) and the material matrix **D** for bending case is given by

$$[\mathbf{D}] = \frac{Et^3}{12(1-\upsilon^2)}\begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & \frac{1-\upsilon}{2} \end{bmatrix} \tag{4.32}$$

The elemental stiffness matrix can be obtained in a way similar to that of in-plane case,

$$[\mathbf{K}]^{eb} = \sum_i \sum_j w_i w_j \mathbf{B}^T \mathbf{D} \mathbf{B} |\mathbf{J}| t \tag{4.33}$$

The FEM code has been developed in FORTRAN and is combined with the GA code in C.

4.3 Shape Realization using Rapid Prototyping

In this section, we will describe the methodology of rapid prototyping the pseudo-intrinsic surface using the FDM 1650 system from Stratasys Inc. Rapid prototyping (RP) is fast becoming one of the essential technologies that allows designers to improve the design of a product and reduced time to physically produce the optimal as well as sub-optimal designs. RP allows designers to build tangible models of their designs quickly and cheaply. In the present work, the optimized shape is produced by the FDM 1650 RP machine to get a feel of the surface design. We will briefly explain the methodology of rapid prototyping of the pseudo-intrinsic surface in the following. For more details about various RP technologies and the description of the FDM 1650 system, please refer to the appendix on rapid prototyping.

The basic steps involved in rapid prototyping using FDM 1650 are:

1. Generating a 3-D CAD model of the surface: The surface is generated in this thesis using Pro-Engineer software by reading the imported blend file (*.ibl file) and then small amount of thickness is added to generate a solid part.

2. Convert the CAD model file into STL format which can be used by the slicing software.

3. Slicing the model: The STL file is read into Stratasys' slicing software-QuickSlice. QuickSlice breaks the model into individual slices, with each slice representing one layer of material. Then QuickSlice generates the tool paths to fill the slices in the form of a SML file.

4. Prototyping the part: The SML file is downloaded to the FDM hardware for manufacturing the part layer by layer.

4.4 Computational Issues

In general, considerable amount of time is consumed by the analysis routine in many of the shape optimization problems. The analysis routine may contain a FEM/BEM simulation before obtaining the required objective function, as is the case with the stiffness of a shell surface in the present study. As discussed in chapter 3, In genetic algorithms the computational time basically depends on the population size and the number of generations for which the GA code is being run. If N is the population size, G is the maximum number of generations, and p_c is the cross over probability then the total time of GA run would be roughly $N \times G \times p_c$. In the FEM analysis of a shell surface, the computational time largely depends on the fineness of the mesh. As the number of elements in each parametric direction increases the total number of dof increases drastically since each node has 6 dof. Thus the order of the stiffness matrix increases and a major portion of the analysis time is spent in solving these Equations. To reduce the solution time bonded matrix form has been used with the Gauss elimination method[7] and it is observed that there is a significant amount of time saving with this method.

The time required to prototype the part using the RP machine will largely depend on the size of the part. It also depends on the thickness of each layer, complexity of the part etc. A flat object of 100x100 mm size with a thickness of 4 mm will take about 9 hours on the FDM 1650 system.

RESULTS AND CONCLUSIONS

5.1 Introduction

In this chapter, we will consider case studies to illustrate the working of the proposed methodology of shape optimal design. Designing a surface for minimum surface area and maximum stiffness are studied. At the end of the chapter, a technical summary of the research work is given. Besides some conclusions about the approach, several suggestions for further work in this direction have been outlined.

Most of the software program has been developed using C language, except for the FEM program for the analysis of the shell surface. FEM codification is done using FORTRAN. The software has been developed on UNIX platform. GA code available in C is used for the optimization of the pseudo-intrinsic surface. Shape synthesis and analysis modules were called by the GA program for the subsequent optimization procedure. Bounds on the design variables are chosen such that a feasible design can be achieved. Different initial random populations are created by supplying a different random seed number. Tournament selection is used for reproduction, and single point crossover is used. The surface is designed between the points (0, 0, 0) and (100, 100, 100) in the XYZ coordinate system.

5.2 Minimum Area Surface

The surface is thought of as an assembly of number of flat trapezoidal elements. The area of each such an element is calculated and summed up to arrive at the total surface area of the

given surface. As mentioned in Section 4.1, this problem is same as the minimum weight design problem when the thickness of the surface is constant throughout.

In order to use genetic algorithms for any optimization problem one has to specify parameters like population size, string length, crossover and mutation probabilities etc. There is no way by which one can fix these GA parameters. Only through some experimentation one can arrive at optimum values of these parameters so that one can get a near optimal solution with less computational effort. Since this work is only to illustrate the application of GAs to the shape optimization problems, we have chosen the frequently used GA parameter values as a reference to start with for the present analysis. It has also been found that with a minor change in any of the above values, the obtained solutions are approximately identical to what we have presented here.

The following values of GA parameters are considered for the present analysis:

Population Size	=	50
Max. No. of Generations	=	25
Total String Length	=	32
Crossover Probability	=	0.8
Mutation Probability	=	0.01
Tournament Size	=	5

String length and bounds of each variable are assumed as shown in Table 5.1.

Table 5.1: String lengths and variable bounds on the Shape Design Variables

Variable	String Length	Variable Bounds	
		Lower	Upper
Sharpness λ_1	6	0.0001	0.005
Sharpness λ_2	6	0.001	0.01
Tangent angle ψ_{10}	5	0.01	30.0
Tangent angle ψ_{11}	5	30.0	70.0
Tangent angle ψ_{20}	5	0.01	10.0
Tangent angle ψ_{21}	5	0.01	10.0

The total string length was taken 32. Since sharpness is relatively more critical design parameter than the tangent slopes, it is decided to code it as a 6-bit string for more accuracy, while tangent angles are coded as 5-bit strings.

Table 5.2 gives a summary of the results obtained with different initial random populations. GA was run for a total of 25 generations in each run. The average fitness after every five generations is given in Table 5.2(a). The last two columns in this table give the fitness of the optimum solution or design and the generation at which it was found. Table 5.2(b) gives the values of SDVs at this best solution.

Table 5.2: Results of the analysis for minimum area surface with different initial random populations

(a) Statistics

GA run	random seed	Average Fitness ($x10^6$)						Optimum sol.	
		0	5	10	15	20	25	fitness	gen.
1	0.123	140	20	80	40	40	20	14305	19
2	0.235	120	40	120	20	140	60	14288	25
3	0.571	160	120	60	140	40	60	14290	15
4	0.395	200	20	40	20	60	20	14290	17
5	0.759	20	100	0.014	20	100	20	14330	25

(b) Shape Design Variables corresponding to the best solution

GA run	Best Solution (SDVs)					
	λ_1	λ_2	ψ_{10}	ψ_{11}	ψ_{20}	ψ_{21}
1	0.00072	0.01	30.0	48.0645	10.0	10.0
2	0.00134	0.01	30.0	46.7742	10.0	10.0
3	0.00258	0.01	30.0	46.7742	10.0	10.0
4	0.00258	0.01	30.0	46.7742	10.0	10.0
5	0.00025	0.01	30.0	50.6450	10.0	10.0

A few things are worth noting from these results. In all the cases four out of the six Shape Design Variables (SDVs) are same in the final optimal design. These four variables are converging towards their corresponding lower or upper variable limits, which means that if we relax these

bounds we may get a flat surface, as expected. Another similarity is that the fitness i.e. the area of the surface at the optimum point is around 14300 units in all the five GA runs. One can observe the fluctuation in the average fitness of the total population at some specific generations. This is due to the fact that we have introduced a small mutation by flipping some of the bits. It should be noted that we have taken the fitness of the infeasible design equal to arbitrarily a large number 999999999.0. The minimum area we get is 100x100 = 10000, in case of a flat surface. It can also be observed that in case 3, the best solution is found at 15th generation itself. There is no further improvement in the optimum solution after this generation. Whereas in GA runs of 2 and 5, the optimum is observed only at the 25th generation, by which one can not say whether the solution has converged or not.

Table 5.3 shows similar results for the same data but with no mutation. It can be seen that the average fitness monotonically decreases towards the optimum value. Only in the initial generations few infeasible designs are created and they will be discarded in the later generation. There will be no possibility of a infeasible design being created afterwards as there is no mutation operator to alter some of the bits randomly. Just after 10 generations there is no improvement in the optimum solution at all.

Table 5.3: Statistics with different initial random populations with no mutation

GA	random	Average Fitness						Optimum sol.	
run	seed	0	5	10	15	20	25	fitness	gen.
1	0.123	$140x10^6$	14328	14317	14316	14316	14316	14316	7
2	0.235	$120x10^6$	14394	14347	14347	14347	14347	14347	8
3	0.571	$120x10^6$	14371	14347	14347	14347	14347	14347	4
4	0.395	$180x10^6$	14370	14368	14368	14368	14368	14368	3
5	0.759	$20x10^6$	14398	14392	14392	14392	14392	14392	2

Rapid Prototyping:

Inorder to get a tangible model of the optimal design, the optimized surface shape is being realized using the RP technique. The surface produced in case of minimum area surface was with the following parameters:

Population Size	=	30
Max. No. of Generations	=	25
Total String Length	=	30
Crossover Probability	=	0.8
Mutation Probability	=	0.01
Tournament Size	=	5
Random seed	=	0.123

The string length and variable bounds of each SDV are given in Table 5.4.

Table 5.4: String lengths and variable bounds on the Shape Design Variables for the rapid prototyped minimum area surface

Variable	String Length	Variable Bounds	
		Lower	Upper
Sharpness λ_1	6	0.0001	0.005
Sharpness λ_2	6	0.001	0.01
Tangent angle ψ_{10}	5	0.01	30.0
Tangent angle ψ_{11}	4	30.0	70.0
Tangent angle ψ_{20}	5	0.01	10.0
Tangent angle ψ_{21}	4	0.01	10.0

Surface area of this surface is 14310.5 units obtained at the 25th generation. The SDVs corresponding to this area are as follows,

Sharpness λ_1	=	0.000411
Sharpness λ_2	=	0.009857
Tangent angle ψ_{10}	=	30.0
Tangent angle ψ_{11}	=	48.67
Tangent angle ψ_{20}	=	10.0
Tangent angle ψ_{21}	=	10.0

Wire frame model of the 3-D surface was created using the AutoCAD software. This surface is exported to the 3-D STUDIO for rendering in order to have a better visualization of the surface. This rendered surface is shown in Figure 5.1.

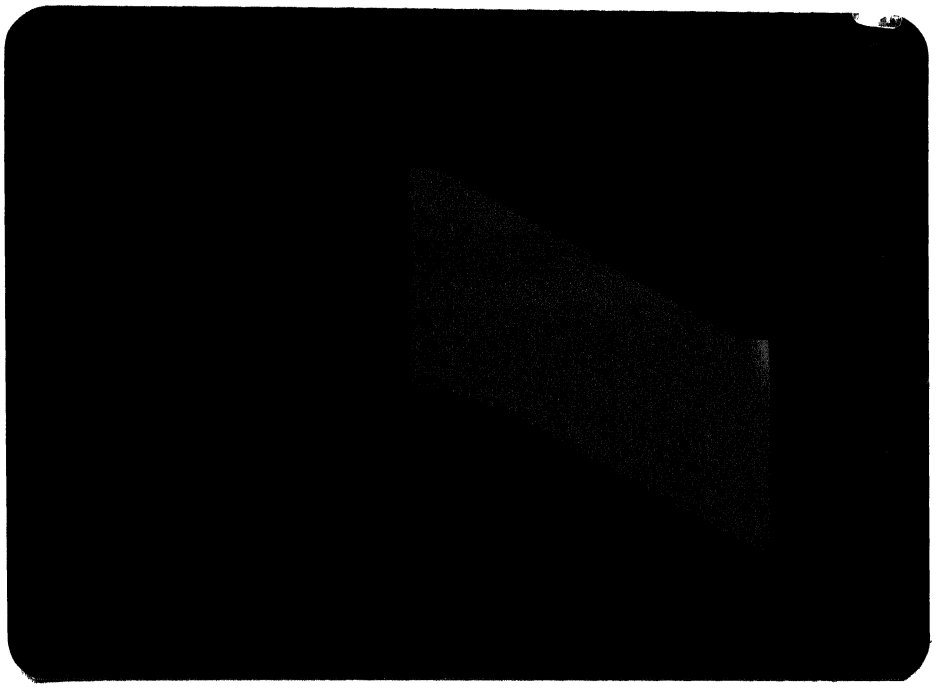


Figure 5.1: Photograph of the surface model of the minimum area surface

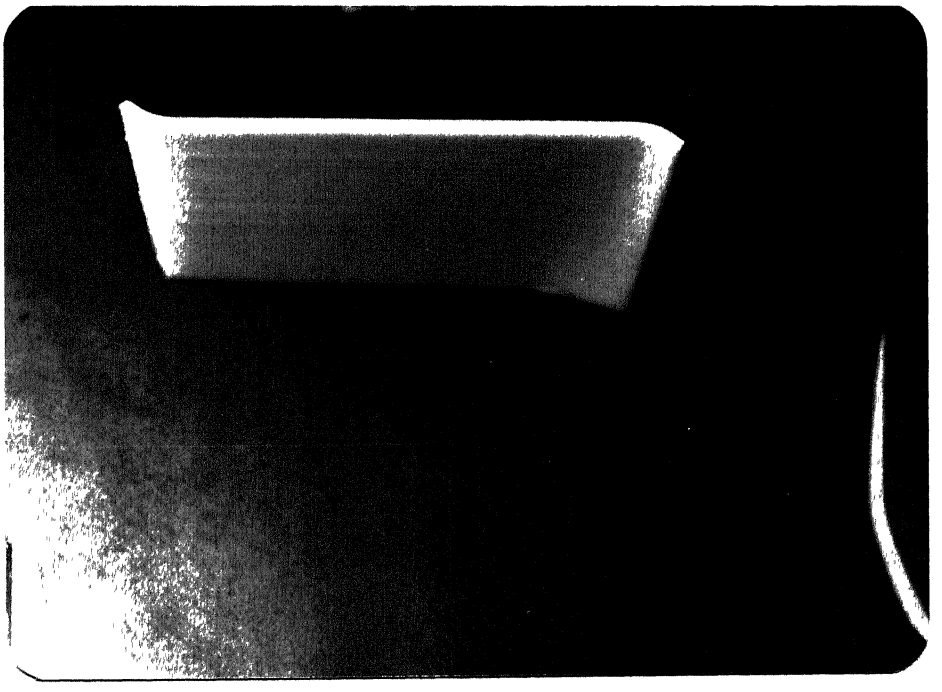


Figure 5.2: Photograph of the rapid prototype of the optimum minimum area surface

As described in Section 4.3, the solid model of this surface is generated using the Pro-engineer software and this file is exported to the QuickSlice software in the form of a STL file. QuickSlice generates the necessary SML file after slicing the model. This SML file is downloaded to the FDM 1650 machine for building the prototype. Also refer appendix for more details on various RP techniques. A photograph of the physical prototype is shown in Figure 5.2.

5.3 Maximum Stiffness Surface

The shell surface is designed for maximum stiffness as follows. As described in Section 4.2, maximizing the stiffness is nothing but minimizing the deflection for a given loading. Using this principle, we have calculated the deflection of each node and the maximum of all these deflections is minimized to get a maximum stiffness surface. It is subjected to a uniform pressure loading in Z-direction with displacement boundary conditions as shown in Figure 5.3. Edges $y = 0$ and $y = 100$ are assumed to be fixed i.e. all the three displacements are equal to zero ($u = v = w = 0$). Edges $x = 0$ and $x = 100$ are left free. Intensity of pressure loading is taken as 100 units. Material properties are as follows:

Thickness	=	1.0 mm
Young's modulus	=	210.0MPa
Poisson's ratio	=	0.3

The surface is divided into 10 elements along the C_1 direction and 20 elements in the C_2 direction. Therefore total number of elements in the finite element mesh are 200. As each element has four nodes, total number of nodes in the finite element model are 221. Since six degrees of freedom are considered per each node, total number of dof in the finite element model are 1326.

Even though the bonded matrix format is used to solve the finite element equations it is found that analysis of a single design takes about one minute time. Because of the time constraints, we have run GA with a smaller population and for lesser number of generations.

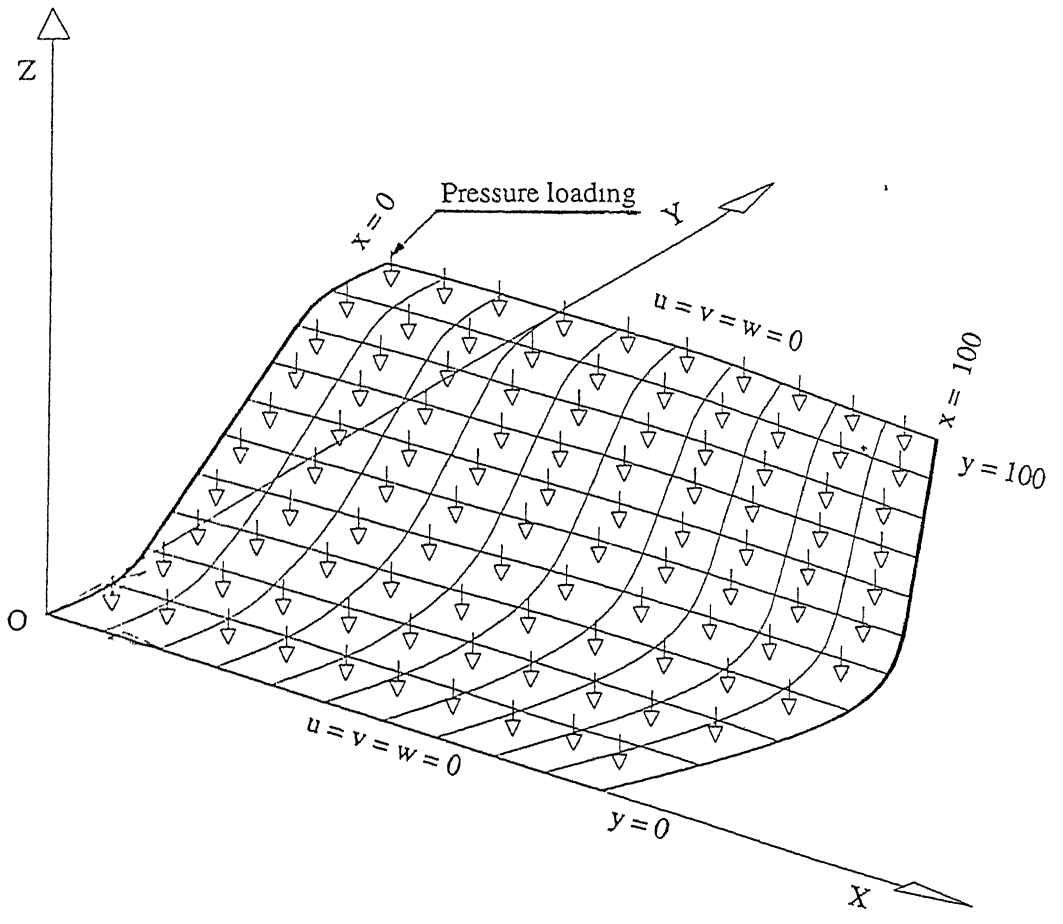


Figure 5.3: Boundary conditions on the shell surface

The following values of GA parameters are considered for the present analysis:

Population Size	=	25
Max. No. of Generations	=	10
Total String Length	=	32
Crossover Probability	=	0.8
Mutation Probability	=	0.01
Tournament Size	=	5

String length and bounds of each variable are assumed similar to the previous case, as given in Table 5.1. Fitness of the infeasible design is taken as 99.99×10^9. The results of five GA runs with different initial random populations for the above input data are depicted in Table 5.5.

Table 5.5: Results of the analysis for maximum stiffness surface with different initial random populations

(a) Statistics

GA run	random seed	Average Fitness			Optimum sol.	
		0	5	10	fitness	gen.
1	0.123	$7.7x10^9$	$3.8x10^9$	5032	3297	5
2	0.235	$19x10^9$	$3.8x10^9$	$3.8x10^9$	3707	2
3	0.571	$23x10^9$	44756	5246	3705	8
4	0.395	$42x10^9$	29580	$3.8x10^9$	3312	2
5	0.759	$3.8x10^9$	7738	5506	3504	4

(b) Shape Design Variables corresponding to the best solution

GA run	Best Solution (SDVs)					
	λ_1	λ_2	ψ_{10}	ψ_{11}	ψ_{20}	ψ_{21}
1	0.00460	0.00328	28.065	53.225	4.1993	10.000
2	0.00144	0.00843	18.391	51.935	8.3887	2.588
3	0.00381	0.00857	12.580	50.645	0.9767	9.033
4	0.00381	0.00657	19.358	50.645	1.2990	3.555
5	0.00246	0.00457	18.391	53.226	0.3323	6.133

Since this is a complex objective function, there is a striking difference between the obtained optimum solutions in both the problems. There is no biasing of the SDVs towards their bounds. It should be noted that the fitness is neither a deflection nor the stiffness because the problem is a minimization of deflection problem, because fitness is evaluated as 1/(1+deflection). Tournament size is taken of 5 is used in all these results. With larger tournament size whole population will be filled up within a few generations, which may lead to local convergence in some problems. However, for the present problems, with a tournament size of 2 or 3 also similar type of results are found.

Rapid prototyping:

Using the procedure described in the case of minimum area surface, surface model and the rapid prototype are created. Rendered surface is shown in Figure 5.4 and the rapid prototype of the maximum stiffness surface is shown in Figure 5.5. This surface is for the optimum design obtained in GA run 1.

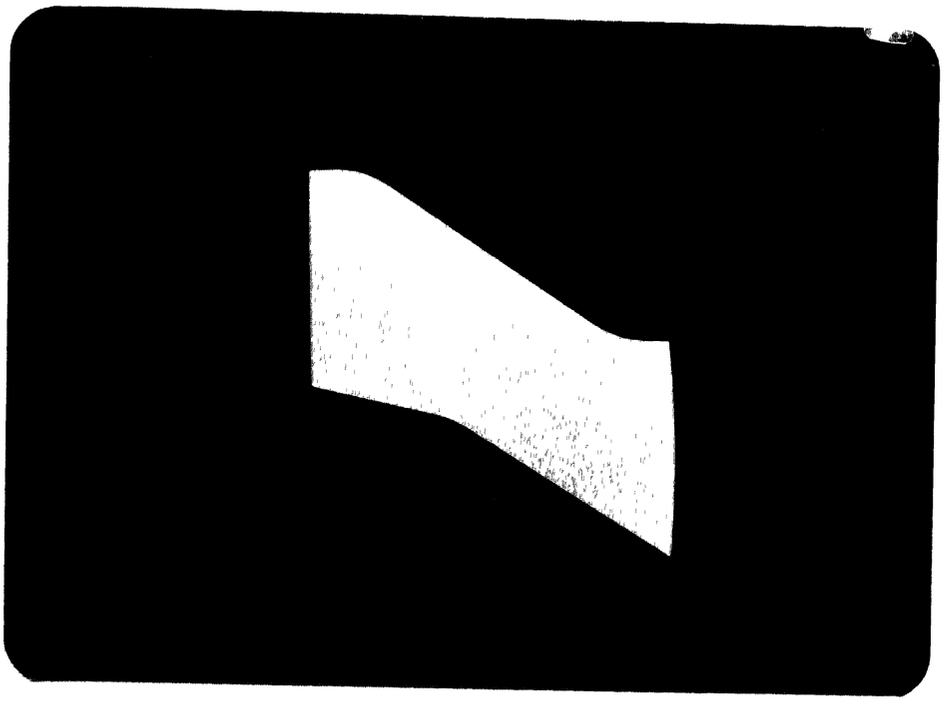


Figure 5.4: Photograph of the surface model of the maximum stiffness surface

Figure 5.2: Photograph of the rapid prototype of the optimum maximum stiffness surface

5.4 Conclusions

5.4.1 Technical Summary

In the present research work, the main focus was on the shape synthesis and optimization using intrinsic geometry and genetic algorithm. Shape optimization is becoming increasingly popular because it often results in a better design with the available resources. Often times it is found that best design also results in an aesthetically pleasing design. We have started with a brief introduction on the importance of shape optimization in improving a design. Applications of surface design in automobile, aerospace, and many other fields have been outlined. The methods adopted in surface design using the Computer Aided Design (CAD) are described. All these methods suffer from the drawback that there is no control on the intrinsic parameters of the curves and surfaces, because these techniques are based on control points. Intrinsic definition involves lesser design variables to control and relates some of the physical parameters with the intrinsic parameters, which allows much flexibility in shape design. Many of the works in shape design using intrinsic geometry are constrained to only 2-D problems. But our approach is designing a 3-D shape which is a novel idea.

We have developed a methodology of shape synthesis in which we define two curves in the intrinsic domain and linearly interpolate them to obtain a ruled surface again in the intrinsic domain. The two curves are selected based on their relatively frequent occurrence in many of the engineering and design problems. This surface is then mapped in to the Cartesian space by using the Simpson's numerical integration scheme. Optimization methodology using GAs is developed. Application of this methodology of shape synthesis and optimization is illustrated by two case studies viz. minimum area surface and maximum stiffness shell surface. A FEM code was developed in FORTRAN for the later case which considers plate bending as well as in-plane loading. Finally use of rapid prototyping in engineering design is depicted by prototyping the optimal design surface using the FDM 1650 RP machine. The software is developed in UNIX environment using C and FORTRAN. Shape synthesis and engineering analysis modules have been combined with the GA code for optimization.

There is no need of generating the finite element mesh separately; since, by changing the number of divisions in either parametric direction one can achieve the desired fineness of mesh. Thus finite element mesh updation is so simple unlike the traditional optimization techniques employing control points concept.

5.4.2 Suggestions for Further Work

Even though we have illustrated the proposed methodology of shape optimization using a fixed shape model (i.e. Figure 2.3(a) for the S-C-S curve and Figure 2.4(b) for the C-S-C curve), the principle can be extended to the other shape models also. This requires adding another variable to the GA coding which enables choosing any of the shape models randomly. This would probably result in a much better optimum design.

The methodology developed in this work can be used for problems like minimum drag or maximum lift surface. Since the pseudo-intrinsic surface is a doubly curved surface, in order to find the drag coefficient one has to solve the full set of Navier-Stokes equations. As the curvature changes continuously from point to point there is no short cut way by which one can evaluate the drag resistance offered by this surface when air or any other fluid flows past it. Because of time limitations this has not been taken up in the present work. In the future work this problem can be explored by developing a finite element program to solve the Navier-Stokes flow equations.

Another possibility is developing a multi-window display system in which the surface can be displayed in intrinsic as well as Cartesian space. This allows the user to visualize how a change in the intrinsic geometry affects the surface shape, in a user-friendly manner.

The concepts developed in this work can be used to design a cantilever beam for minimum weight under different constraints like stress, frequency etc. This surface can be taken as top surface of the beam and using the symmetry we can design the complete beam by developing a finite element code. The computer time requirements are fairly large in case of FEA. Most of the time is spent during the FE analysis, hence equation solvers like sky-solver method, wave front minimization techniques can be employed.

In order to make sure that the solution obtained by using the genetic algorithm is really an optimal one, we can take a problem which has a known solution and solve it by using some traditional optimization method as well as GA. Compare these results. If the results obtained with GA are comparable to those obtained with the traditional technique and the analytical solution, we can conclude that GA can be effectively used to solve other shape optimization problems also.

REFERENCES

[1] Adams, J. A., 1975, "The Intrinsic Method for Curve Definition", *Computer Aided Design*, Vol. 7, No. 4, pp. 243-249.

[2] Arora, J. S., 1989, *Introduction to Optimum Design*, McGraw-Hill, New York.

[3] Braibant, V., and Fleury, C., 1984, "Shape Optimal Design using B-splines", *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, pp. 247-267.

[4] Chaturvedi, D., Deb, K., and Chakrabarti, S. K., 1995, "Structural Optimization using Real-Coded Genetic Algorithm", *Proceedings of the Symposium on Genetic Algorithms*, pp. 73-81.

[5] Cheu, T., Wang, B. P., and Chen, T., 1989, "Design Optimization of Gas Turbine Blades with Geometry and Natural Frequency Constraints", *Computers & Structures*, Vol. 32, No. 1, pp. 113-117.

[6] Chu, P. C., and Beasley J. E., 1997, "Constraint Handling in Genetic Algorithms: the Set Partitioning Problem", *in press*.

[7] Cook, R. D., 1984, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, New York.

[8] Deb, K., 1995, *Optimization Methods for Engineering Design: Algorithms and Examples*, Prentice-Hall, New Delhi.

[9] Deb, K., 1996, "Genetic Algorithms for Engineering Design Optimization", A collection of papers prepared for the course ME 767 - *Evolutionary Algorithms in Search and Optimization*, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur.

[10] Deb, K., 1991, "Optimal Design of a Welded Beam via Genetic Algorithms", *AIAA Journal*, Vol. 29, No. 11, pp. 2013-2015.

[11] Deb, K., 1996, "An Overview of Traditional Optimization Methods", A report prepared for the course *Traditional and AI-based Optimization Methods in Process Metallurgy*, Volta Redonda, Brazil.

[12] Ding, Y., 1987, "Shape Optimization of Two-Dimensional Elastic Structures with Optimal thickness for Fixed parts", *Computers & Structures*, Vol. 27, No. 6, pp. 729-743.

[13] Ding, Y., 1986, "Shape Optimization of Structures: A Literature Survey", *Computers & Structures*, Vol. 24, No. 6, pp. 985-1004.

[14] Ebrahimi, N. D., 1988, "Optimum Design of Flywheels", *Computers & Structures*, Vol. 29, No. 4, pp. 681-686.

[15] Espiga, F., Gracia, L., and Doblare, M., 1989, "Shape Optimization of Elastic Homogeneous 2D Bodies by the Boundary Element Method", *Computers & Structures*, Vol. 33, No. 5, pp. 1233-1241.

[16] Faux, I. D., and Pratt, M. J., 1983, *Computational Geometry for Design and Manufacture*, Ellis Horwood Publishers, Chichester.

[17] Ghoshtasby, A., 1995, "Geometric Modelling using Rational Gaussian Curves and Surfaces", *Computer Aided Design*, Vol. 27, No. 5, pp. 363-375.

[18] Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York.

[19] Goldberg, D. E., 1986, *A Simple Genetic Algorithm Code*, C Version, v1.1, University of Alabama.

[20] Griffiths, D. V., 1994, "Stiffness Matrix of the Four-node Quadrilateral Element in Closed Form", *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 1027-1038.

[21] Grigson, A., 1994, "Model Making", *Manufacturing Engineer*, Vol. , pp. 172-178.

[22] Haftka, R. T., and Grandhi, R. V., 1986, "Structural Shape Optimization–A survey", *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, pp. 91-106.

[23] Holland, J., H.,1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor MI.

[24] Hemp, W. S., 1973, *Optimum Structures*, Clarendon.

[25] Imam, M. H., 1982, "Three Dimensional Shape Optimization", *International Journal for Numerical Methods in Engineering*, Vol. 18, pp. 661-673.

[26] Jenkins, W. M., 1991, "Towards Structural Optimization via the Genetic Algorithm", *Computers & Structures*, Vol. 40, No. 5, pp. 1321-1327.

[27] Mortenson, M. E., 1985, *Geometric modelling*, Wiley, New York.

[28] Nutbourne, A. W., and Martin, R. R., 1988, *Differential Geometry Applied to Curve and Surface Design*, Vol. 1, Ellis Horwood Limited, Chichester.

[29] Pal, T. K., and Nutbourne, A. W., 1977, "Two Dimensional Curve Synthesis using Linear Curvature Elements", *Computer Aided Design*, Vol. 9, No. 2, pp. 121-134.

[30] Pham, D. T., and Gault, R. S., 1996, "Solid Ideas", *Manufacturing Engineer*, Vol. 75, No. 5, pp. 239-243.

[31] Pham, D. T., and Gault, R. S., 1996, "Modelling for all Occasions", *Manufacturing Engineer*, Vol. 75, No. 6, pp. 289-293.

[32] Qin, H., and Terzopoulos, D., 1995, "Dynamic NURBS Swung Surfaces for Physics based Shape Design", *Computer Aided Design*, Vol. 27, No. 2, pp. 111-127.

[33] Rao, S. S., 1991, *Optimization Theory and Applications*, Wiley Eastern Limited, New Delhi.

[34] Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M., 1989, *Engineering Optimization Methods and Applications*, John Wiley & Sons, New York.

[35] Riche, R. L., and Haftka, R. T., 1993, "Optimization of Laminate Stacking Sequence for Buckling Load Maximization by Genetic Algorithm", *AIAA Journal*, Vol. 31, No. 5, pp. 951-956.

[36] Rogers, D. F., and Adams, J. A., 1989, *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York.

[37] Shyy, Y. K., Fleury, C., and Izadopanah, K., 1988, "Shape Optimal Design using Higher Order Elements", *Computer Methods in Applied Mechanics and Mechanical Engineering*", Vol. 71, pp. 99-116.

[38] Srinivas, Y. L., Vinod Kumar, K. P., and Dutta, D., 1996, "Surface Design using Cyclide Patches", *Computer Aided Design*, Vol. 28, No. 4, pp. 263-276.

[39] Srinivasa Rao, N., 1995, "Design of Trajectories using Intrinsic Geometry", *M. Tech Thesis Submitted to the Department of Mechanical Engineering*, Indian Institute of Technology, Kanpur, India.

[40] Struik, D. J., 1961, *Lectures on Classical Differential Geometry*, Addison Wesley Press, Inc., Cambridge 42, Mass.

[41] Tavakkoli, S., 1991, "Shape Design Using Intrinsic Geometry", *Ph. D. Dissertation Submitted to the Department of Mechanical Engineering*, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA.

[42] Tavakkoli, S., and Dhande, S. G., 1991, "Shape Synthesis and Optimization using Intrinsic Geometry", *ASME Journal of Mechanical Design*, Vol. 113, No. 4, pp. 379-386.

[43] Timoshenko, S. P., 1983, *History of Strength of Materials*, Dover Publications, New York.

[44] Vanderplaats, G. N., 1984, *Numerical Optimization Methods for Engineering Design*, McGraw-Hill, New York.

[45] Venkatesh, J., 1994, "Shape Optimization using Intrinsic Geometry and Boundary Element Method", *Ph. D. Dissertation Submitted to the Department of Mechanical Engineering*, Indian Institute of Technology, Kanpur, India.

[46] Xinzi, Y., Jackson, T. R., and Patrikalakis, N. M., 1996, "Geometric Design of Functional Surfaces", *Computer Aided Design*, Vol. 29, No. 9, pp. 741-752.

[47] Zienkiewicz, O. C., and Taylor, R. L., 1991, *The Finite Element Method*, Vol. 2, McGraw-Hill, New York.

RAPID PROTOTYPING

1 An Overview of RP Technologies

Rapid prototyping (RP) is fast becoming one of the essential technologies that allows manufacturers to improve product quality and reduce both time to market and the cost of new lines, as part of a concurrent engineering strategy. Rapid prototyping technologies can be divided into two groups ([21], [30], and [31]):

- Virtual rapid prototyping
- Physical rapid prototyping

Virtual prototyping (soft/CAD-based) implies visualizing and testing of CAD models on a computer before they are physically created. Virtual prototyping is accomplished by running a computer model through iterative dynamic simulations before making a physical prototype. Physical or real rapid prototyping (which will be referred as *rapid prototyping,* for brevity, hereafter) allows designers to build tangible models of their designs quickly and cheaply. This encourages experimentation, and flag errors in the fit or assembly of the part. It has been claimed that RP model can cut new product costs by up to 70% and the time to market by 90% [30].

Rapid prototyping, or free form fabrication as the Americans prefer to call it, is about building a prototype from a CAD drawing in a matter of hours or days rather than months traditionally taken. RP technologies may be divided broadly into those involving material accretion and those involving material removal. Material-accretion technologies can be further

divided into liquid-based, powder-based, and solid sheet based, depending on the state of the prototype material before part formation. Liquid-based technologies may involve the solidification of a resin on contact with a laser, solidifying an electrosetting fluid, or melting and setting the prototyping material. Processes using powders compound them either with a laser or by the selective application of binding agents. Processes using solid sheets bond them either with a laser or with an adhesive. Figure 1 shows Kruth's classification, which has been adapted to include new technologies.

All the RP processes that will be reviewed here take input from a 3D solid CAD model, usually as slices. The 3D CAD model is tessellated and exported as an STL file. The model is then split into thin slices and the slices are sent to the RP machine for the production of the final physical product. By convention, the data slices are said to be in the x-y plane and the part is built in the z-direction. The number and position of the supports partly depends on the build direction chosen. Part orientation will also influence the final prototype build time and the surface finish of critical areas. The following are the currently available RP methods that are in commercial use, with several others in various stages of development:

- Steriolithography
- Solid ground curing
- Selective laser sintering
- Fused deposition modelling
- Laminated-object manufacturing
- Ballistic particle manufacturing
- Three-dimensional printing

Steriolithography (SL) was the first RP technology to be developed and currently, it is probably the most widely used one. This uses a photosensitive monomer resin (usually acrylic or epoxy resin) that forms a polymer and solidifies, on the surface, when exposed to ultraviolet light. SL produces a surface finish that is comparable to that of NC milling; it is a well proven system

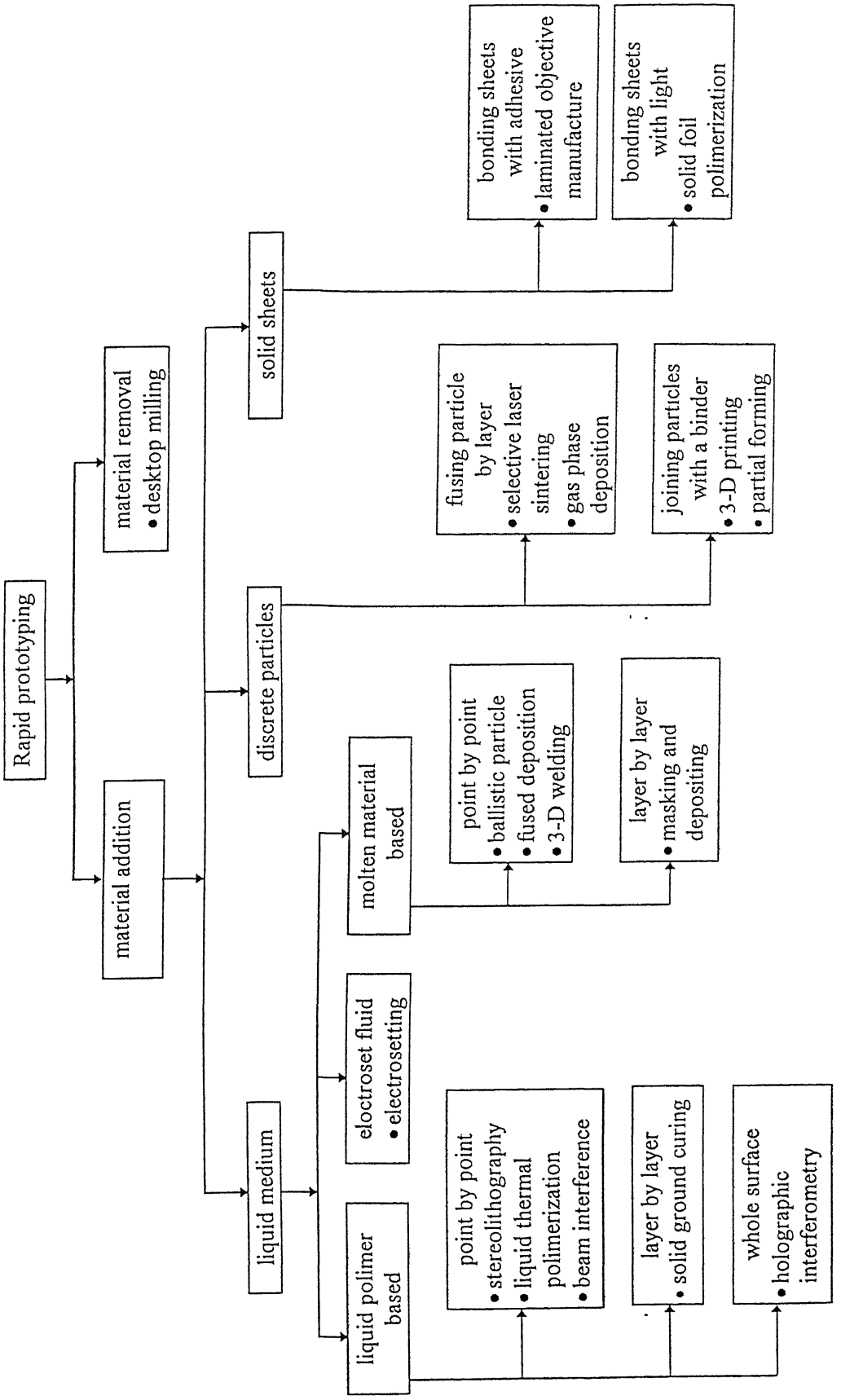


Figure 1: Classification of Rapid-prototyping methods

with over 500 machines in use worldwide, and is reasonably fast and accurate. As an economy of scale several parts can be built at once. On the other hand, the material is expensive, smelly, and toxic, and must be shielded from light to avoid premature polymerization; there is also a limited choice of resins.

Solid ground curing (SGC) also uses photo-polymerizing resins. An image of the individual layer is made on the glass plate and then placed over a liquid resin tank. The exposed parts of the resin are cured using a burst of ultraviolet light, and the uncured resin is removed from the protected areas by an aerodynamic wiper and replaced by wax. This wax solidifies to provide a supporting structure for the growing model. An advantage of this process is that a whole layer is solidified at once, making the whole process relatively fast. All the resin within the layer is completely cured, so parts may be stronger than hatched SL prototypes, and no postcuring is needed. Supports are unnecessary because wax is used to fill the gaps in the resin and to brace the part. On the other hand, SGC is noisy, massive and needs to be constantly attended.

Selective laser sintering (SLS), uses a laser to build up a shape with powder, rather than liquid resin. Any material that can be pulverized and melted may be used. e.g., Nylon and its composites, wax, sand, and polycarbonates etc. Each layer of powder is fused together using CO_2 laser to trace out the shape. A new layer of powder is then spread over the top by a roller and the next layer is formed. Each layer is fused to the last. SLS also doesn't need supporting structure as it is supported by unused powder in the container and no postcuring is necessary, except with ceramics. Materials used as powder are relatively cheaper than the resins for SL, are non-toxic, safe and may be sintered with relatively low-powered lasers. But prototypes made using this method have poor mechanical properties, since they are porous in general. An inert nitrogen atmosphere is required to sinter the material.

Fused deposition modelling (FDM) involves the extrusion of wax or thermoplastics just above the melting point using a computer controlled deposition head (nozzle). The head is fed by a spool of non-toxic thermoplastic filament material of 0.5" diameter which solidifies as it is directed into place by the x-y controlled head, creating the required model. FDM uses two

nozzles, one for the part material and one for the support material, which is cheaper and breaks away from the prototype without damaging its surface. It is also possible to create horizontal supports to minimize material usage and build time. FDM can be viewed as a desktop prototyping technology, suitable for use in a design office, since the materials used are cheap, clean and safe. There is also a large range of colors and materials available, such as investment-casting wax, ABS plastic and elastomers. Parts made by this method have a high stability since they are not higroscopic. However, surface finish is not good because of the resolution of the process, which is dictated by the filament thickness. It may be unable to produce small holes in vertical sections.

Laminated object manufacture (LOM) is a solid technology, wherein the build material is applied to the part from a roll, then bonded to the previous layers a hot roller, which activates a heat-sensitive adhesive. The contour of each layer is cut with a CO_2 laser, carefully modulated to penetrate exactly one layer thickness. Unwanted material is trimmed into rectangles to facilitate its later removal, but remains in place for support. LOM is cheap because paper can be used as the base material. It is 5-10 times faster than the other processes. One drawback is the need to prise the finished parts off the machine table, which can damage surface finish. It is also hart to make hollow parts because of the difficulty of removing the core, and there are serious problems with undercuts and re-entrant features. Warping of the laminations due to the heat from the laser is also seen.

In **ballistic particle manufacture**, a stream of molten material is ejected from a nozzle; the material separates into droplets, which hit the substrate and immediately cold weld to form the part. BPM is cheap and environmentally benign, but it is restrictive in the small range of commercial materials available to construct the prototypes. From the systems available, either speed or accuracy is possible, but not both.

Three-dimensional printing is also a powder-based process using a separate binder: layers of powder are applied to a substrate, stabilized by misting with water, then selectively joined using a binder sprayed through a nozzle. Once the part is completed, it is heated to set the binder; the excess powder, which was supporting the part, is then rinsed off. The part is fired then to sinter it. To increase the density of the part, it is possible to press the 'green' part before this

final firing. After firing, the part may be dipped in binder to strengthen it; there is no state change involved, so distortion is reduced. Parts do not need supports to brace the overhanging features, but they do need to include a hole so that excess powder can be removed. But, the final parts may be fragile and porous, and it can be hard to remove the unbound powder from any cavities. Further, as the layers are raster scanned by the printhead, stair-stepping effect occurs in the x-y plane as well as the build direction. The materials employed may be metal or ceramic powders, or metal-ceramic composites with colloidal silica or polymeric binders.

As we have seen in this overview, the RP technology is far from mature; many advances are in the pipeline, improving process accuracy, the choice, strength and cost of materials and the overall dimensional stability of the part. Other RP systems under research include [21] - shape melting technology, photochemical machining, and light sculpting etc. In the next section, we will discuss the FDM-1650 system from Stratasys inc.

2 FDM 1650 System

There are four main parts in the Stratasys modeling system:

- Slicing software
- Computer workstation
- FDM 1650 modeler
- Modelling materials

The FDM 1650 uses Fused Deposition Modelling (FDM) to turn computer-aided design (CAD) geometry into models that can be used for design reviews, manufacturability studies, investment casting patterns, and marketing. The FDM 1650 features Break Away Support System (BASS), allowing the designer to create models with greater speed and precision. The support tip extrudes a material that supports any overhanging portions of the model's geometry. When the model is completed, the support is easily broken off leaving behind the final part. BASS system, not only produces a better quality finished product than a part made without a BASS system, but the supports come off easily and cleanly. The FDM 1650 is a bench-top unit standing forty-five

and one half inches high with a 26x36 inches footprint. Because, it requires no exhaust hood or other special facilities, it can be placed next to a designer's CAD workstation.

Each CAD file is converted to an STL format. The STL file is read into Stratasys' slicing software called QuickSlice. QuikSlice breaks the model into individual slices, with each slice representing one layer of material. QuickSlice then generates the tool paths to fill the slices. These tool paths form the Stratasys Modelling Language (SML) file. This SML file is downloaded to the FDM hardware for prototyping the part.

In the FDM hardware, the FDM head moves in two horizontal axes across a foundation and deposits a layer of material for each slice. The material is heated by the FDM head, so it comes out in semi liquid state. The successive layers fuse together and solidify to build up an accurate three-dimensional model of the design. Overall tolerance is ± 0.005″ in the X, Y, and Z axes. Actual results depend on the model.

FDM 1650 is capable of using a variety of inert, non-toxic materials. Currently, Investment casting wax and P400 plastic - ABS plastic are widely used. Each material comes wound on a spool in the form of a filament of approximately 0.07″ in diameter, so it is easy to load as well as to store. Investment casting wax allows the operator to create a wax pattern from the STL file. Forming and dewaxing a shell mould is done rapidly by using normal investment casting procedures. No special burnout is required. P400 ABS plastic is an acrylonitrile-butadiene-styrene based material which produces sturdy prototypes.

In summary, with the FDM 1650 system models can be created in;

- modelling envelop sizes of upto 9.4x10x10 inches
- a variety of wall thicknesses ranging from 0.012 to 0.08 inches
- slice resolutions ranging from 0.005 to 0.02 inches
- investment casting wax and ABS plastic